



GumNaaM_Helpers



Mega File for MiD TerM

All Files in ONE FILE

(Handouts_waqar_moaz_junaid_vu toper_orange_monkey)

Alhamdulillah GumNaaM_Helpers Bs..IT/CS ky Tamam Semesters ki tamam Books ki help krty hn.

Group Join krty hi description Read krna na bholyn

Semester Name	Group Link
Bs..IT/CS...smstr:1	https://chat.whatsapp.com/Ciya2X0DI0I8iQ14m0VNNj
Bs..IT/CS...smstr:2	https://chat.whatsapp.com/Kwh85sRK02z1Qk5B7aMR19
Bs..IT/CS...smstr:3	https://chat.whatsapp.com/ljlpdzi8pIA4IQVNUMmNYE
Bs..IT/CS...smstr:4	https://chat.whatsapp.com/CiGpHsSftir0oyhM47WfKh
Bs..IT/CS...smstr:5	https://chat.whatsapp.com/Hf7NWhdW06uCW7Mbt1n4Db
Bs..IT/CS...smstr:6	https://chat.whatsapp.com/DyAoT7VFkwy8DuvryDqm2L
Bs..IT/CS...smstr:7	https://chat.whatsapp.com/ENiyfIDRWatJFPbXCUGYNN
Bs..IT/CS...smstr:8	https://chat.whatsapp.com/Bnkk4n7utpdEuXazHLxRcA

...Paid Service Available for LMS Activities...

Only for those students who are so busy in office or work or job that they can't watch their LMS due to their Busyness

FOR MORE INFO. CONTACT US ONLY WHAT'S APP 0302-1500025

Design and Analysis of Algorithms

Sohail Aslam

January 2004

Contents

1	Introduction	7
1.1	Origin of word: <i>Algorithm</i>	7
1.2	Algorithm: Informal Definition	7
1.3	Algorithms, Programming	7
1.4	Implementation Issues	8
1.5	Course in Review	9
1.6	Analyzing Algorithms	9
1.7	Model of Computation	10
1.8	Example: 2-dimension maxima	10
1.9	Brute-Force Algorithm	11
1.10	Running Time Analysis	13
1.10.1	Analysis of the brute-force maxima algorithm.	14
1.11	Analysis: A Harder Example	16
1.11.1	2-dimension Maxima Revisited	17
1.11.2	Plane-sweep Algorithm	18
1.11.3	Analysis of Plane-sweep Algorithm	21
1.11.4	Comparison of Brute-force and Plane sweep algorithms	21
2	Asymptotic Notation	23
3	Divide and Conquer Strategy	27
3.1	Merge Sort	27
3.1.1	Analysis of Merge Sort	29
3.1.2	The Iteration Method for Solving Recurrence Relations	31
3.1.3	Visualizing Recurrences Using the Recursion Tree	31

3.1.4	A Messier Example	32
3.2	Selection Problem	34
3.2.1	Sieve Technique	34
3.2.2	Applying the Sieve to Selection	35
3.2.3	Selection Algorithm	35
3.2.4	Analysis of Selection	36
4	Sorting	39
4.1	Slow Sorting Algorithms	39
4.2	Sorting in $O(n \log n)$ time	40
4.2.1	Heaps	40
4.2.2	Heapsort Algorithm	41
4.2.3	Heapify Procedure	43
4.2.4	Analysis of Heapify	43
4.2.5	BuildHeap	43
4.2.6	Analysis of BuildHeap	44
4.2.7	Analysis of Heapsort	46
4.3	Quicksort	46
4.3.1	Partition Algorithm	46
4.3.2	Quick Sort Example	47
4.3.3	Analysis of Quicksort	49
4.3.4	Worst Case Analysis of Quick Sort	49
4.3.5	Average-case Analysis of Quicksort	50
4.4	In-place, Stable Sorting	54
4.5	Lower Bounds for Sorting	54
5	Linear Time Sorting	57
5.1	Counting Sort	57
5.2	Bucket or Bin Sort	69
5.3	Radix Sort	71
6	Dynamic Programming	73
6.1	Fibonacci Sequence	73

6.2	Dynamic Programming	75
6.3	Edit Distance	75
6.3.1	Edit Distance: Applications	76
6.3.2	Edit Distance Algorithm	77
6.3.3	Edit Distance: Dynamic Programming Algorithm	77
6.3.4	Analysis of DP Edit Distance	84
6.4	Chain Matrix Multiply	84
6.4.1	Chain Matrix Multiplication-Dynamic Programming Formulation	85
6.5	0/1 Knapsack Problem	91
6.5.1	0/1 Knapsack Problem: Dynamic Programming Approach	93
7	Greedy Algorithms	97
7.1	Example: Counting Money	97
7.1.1	Making Change: Dynamic Programming Solution	98
7.1.2	Complexity of Coin Change Algorithm	99
7.2	Greedy Algorithm: Huffman Encoding	99
7.2.1	Huffman Encoding Algorithm	100
7.2.2	Huffman Encoding: Correctness	102
7.3	Activity Selection	105
7.3.1	Correctness of Greedy Activity Selection	107
7.4	Fractional Knapsack Problem	109
8	Graphs	113
8.1	Graph Traversal	116
8.1.1	Breadth-first Search	117
8.1.2	Depth-first Search	119
8.1.3	Generic Graph Traversal Algorithm	119
8.1.4	DFS - Timestamp Structure	125
8.1.5	DFS - Cycles	130
8.2	Precedence Constraint Graph	131
8.3	Topological Sort	133
8.4	Strong Components	135
8.4.1	Strong Components and DFS	137

8.5	Minimum Spanning Trees	142
8.5.1	Computing MST: Generic Approach	143
8.5.2	Greedy MST	144
8.5.3	Kruskal's Algorithm	147
8.5.4	Prim's Algorithm	149
8.6	Shortest Paths	153
8.6.1	Dijkstra's Algorithm	154
8.6.2	Correctness of Dijkstra's Algorithm	158
8.6.3	Bellman-Ford Algorithm	159
8.6.4	Correctness of Bellman-Ford	160
8.6.5	Floyd-Warshall Algorithm	161
9	Complexity Theory	169
9.1	Decision Problems	170
9.2	Complexity Classes	170
9.3	Polynomial Time Verification	171
9.4	The Class NP	172
9.5	Reductions	173
9.6	Polynomial Time Reduction	177
9.7	NP-Completeness	177
9.8	Boolean Satisfiability Problem: Cook's Theorem	178
9.9	Coping with NP-Completeness	184

Chapter 1

Introduction

1.1 Origin of word: *Algorithm*

The word *Algorithm* comes from the name of the muslim author *Abu Ja'far Mohammad ibn Musa al-Khowarizmi*. He was born in the eighth century at Khwarizm (Kheva), a town south of river Oxus in present Uzbekistan. Uzbekistan, a Muslim country for over a thousand years, was taken over by the Russians in 1873.

His year of birth is not known exactly. Al-Khwarizmi parents migrated to a place south of Baghdad when he was a child. It has been established from his contributions that he flourished under Khalifah Al-Mamun at Baghdad during 813 to 833 C.E. Al-Khwarizmi died around 840 C.E.

Much of al-Khwarizmi's work was written in a book titled *al Kitab al-mukhtasar fi hisab al-jabr wa'l-muqabalah* (The Compendious Book on Calculation by Completion and Balancing). It is from the titles of these writings and his name that the words *algebra* and *algorithm* are derived. As a result of his work, al-Khwarizmi is regarded as the most outstanding mathematician of his time

1.2 Algorithm: Informal Definition

An algorithm is any well-defined computational procedure that takes some values, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into output.

1.3 Algorithms, Programming

A good understanding of algorithms is essential for a good understanding of the most basic element of computer science: programming. Unlike a program, an algorithm is a mathematical entity, which is independent of a specific programming language, machine, or compiler. Thus, in some sense, algorithm

design is all about the mathematical theory behind the design of good programs.

Why study algorithm design? There are many facets to good program design. Good algorithm design is one of them (and an important one). To be really complete algorithm designer, it is important to be aware of programming and machine issues as well. In any important programming project there are two major types of issues, macro issues and micro issues.

Macro issues involve elements such as how does one coordinate the efforts of many programmers working on a single piece of software, and how does one establish that a complex programming system satisfies its various requirements. These macro issues are the primary subject of courses on software engineering.

A great deal of the programming effort on most complex software systems consists of elements whose programming is fairly mundane (input and output, data conversion, error checking, report generation). However, there is often a small critical portion of the software, which may involve only tens to hundreds of lines of code, but where the great majority of computational time is spent. (Or as the old adage goes: 80% of the execution time takes place in 20% of the code.) The micro issues in programming involve how best to deal with these small critical sections.

It may be very important for the success of the overall project that these sections of code be written in the most efficient manner possible. An unfortunately common approach to this problem is to first design an inefficient algorithm and data structure to solve the problem, and then take this poor design and attempt to fine-tune its performance by applying clever coding tricks or by implementing it on the most expensive and fastest machines around to boost performance as much as possible. The problem is that if the underlying design is bad, then often no amount of fine-tuning is going to make a substantial difference.

Before you implement, first be sure you have a good design. This course is all about how to design good algorithms. Because the lesson cannot be taught in just one course, there are a number of companion courses that are important as well. CS301 deals with how to design good data structures. This is not really an independent issue, because most of the fastest algorithms are fast because they use fast data structures, and vice versa. In fact, many of the courses in the computer science program deal with efficient algorithms and data structures, but just as they apply to various applications: compilers, operating systems, databases, artificial intelligence, computer graphics and vision, etc. Thus, a good understanding of algorithm design is a central element to a good understanding of computer science and good programming.

1.4 Implementation Issues

One of the elements that we will focus on in this course is to try to study algorithms as pure mathematical objects, and so ignore issues such as programming language, machine, and operating system. This has the advantage of clearing away the messy details that affect implementation. But these details may be very important.

For example, an important fact of current processor technology is that of locality of reference. Frequently accessed data can be stored in registers or cache memory. Our mathematical analysis will usually ignore these issues. But a good algorithm designer can work within the realm of mathematics, but still keep an

open eye to implementation issues down the line that will be important for final implementation. For example, we will study three fast sorting algorithms this semester, heap-sort, merge-sort, and quick-sort. From our mathematical analysis, all have equal running times. However, among the three (barring any extra considerations) quick sort is the fastest on virtually all modern machines. Why? It is the best from the perspective of locality of reference. However, the difference is typically small (perhaps 10-20% difference in running time).

Thus this course is not the last word in good program design, and in fact it is perhaps more accurately just the first word in good program design. The overall strategy that I would suggest to any programming would be to first come up with a few good designs from a mathematical and algorithmic perspective. Next prune this selection by consideration of practical matters (like locality of reference). Finally prototype (that is, do test implementations) a few of the best designs and run them on data sets that will arise in your application for the final fine-tuning. Also, be sure to use whatever development tools that you have, such as profilers (programs which pin-point the sections of the code that are responsible for most of the running time).

1.5 Course in Review

This course will consist of four major sections. The first is on the mathematical tools necessary for the analysis of algorithms. This will focus on asymptotics, summations, recurrences. The second element will deal with one particularly important algorithmic problem: sorting a list of numbers. We will show a number of different strategies for sorting, and use this problem as a case-study in different techniques for designing and analyzing algorithms.

The final third of the course will deal with a collection of various algorithmic problems and solution techniques. Finally we will close this last third with a very brief introduction to the theory of NP-completeness. NP-complete problem are those for which no efficient algorithms are known, but no one knows for sure whether efficient solutions might exist.

1.6 Analyzing Algorithms

In order to design good algorithms, we must first agree the criteria for measuring algorithms. The emphasis in this course will be on the design of efficient algorithm, and hence we will measure algorithms in terms of the amount of computational resources that the algorithm requires. These resources include mostly running time and memory. Depending on the application, there may be other elements that are taken into account, such as the number disk accesses in a database program or the communication bandwidth in a networking application.

In practice there are many issues that need to be considered in the design algorithms. These include issues such as the ease of debugging and maintaining the final software through its life-cycle. Also, one of the luxuries we will have in this course is to be able to assume that we are given a clean, fully-specified mathematical description of the computational problem. In practice, this is often not the case, and the algorithm must be designed subject to only partial knowledge of the final specifications. Thus, in practice

it is often necessary to design algorithms that are simple, and easily modified if problem parameters and specifications are slightly modified. Fortunately, most of the algorithms that we will discuss in this class are quite simple, and are easy to modify subject to small problem variations.

1.7 Model of Computation

Another goal that we will have in this course is that our analysis be as independent as possible of the variations in machine, operating system, compiler, or programming language. Unlike programs, algorithms to be understood primarily by people (i.e. programmers) and not machines. This gives us quite a bit of flexibility in how we present our algorithms, and many low-level details may be omitted (since it will be the job of the programmer who implements the algorithm to fill them in).

But, in order to say anything meaningful about our algorithms, it will be important for us to settle on a mathematical model of computation. Ideally this model should be a reasonable abstraction of a standard generic single-processor machine. We call this model a *random access machine* or RAM.

A RAM is an idealized machine with an infinitely large random-access memory. Instructions are executed one-by-one (there is no parallelism). Each instruction involves performing some basic operation on two values in the machine's memory (which might be characters or integers; let's avoid floating point for now). Basic operations include things like assigning a value to a variable, computing any basic arithmetic operation (+, -, \times , integer division) on integer values of any size, performing any comparison (e.g. $x \leq 5$) or boolean operations, accessing an element of an array (e.g. $A[10]$). We assume that each basic operation takes the same constant time to execute.

This model seems to do a good job of describing the computational power of most modern (nonparallel) machines. It does not model some elements, such as efficiency due to locality of reference, as described in the previous lecture. There are some "loop-holes" (or hidden ways of subverting the rules) to beware of. For example, the model would allow you to add two numbers that contain a billion digits in constant time. Thus, it is theoretically possible to derive nonsensical results in the form of efficient RAM programs that cannot be implemented efficiently on any machine. Nonetheless, the RAM model seems to be fairly sound, and has done a good job of modelling typical machine technology since the early 60's.

1.8 Example: 2-dimension maxima

Let us do an example that illustrates how we analyze algorithms. Suppose you want to buy a car. You want the pick the fastest car. But fast cars are expensive; you want the cheapest. You cannot decide which is more important: speed or price. Definitely do not want a car if there is another that is both faster and cheaper. We say that the fast, cheap car *dominates* the slow, expensive car relative to your selection criteria. So, given a collection of cars, we want to list those cars that are not dominated by any other.

Here is how we might model this as a formal problem.

- Let a point p in 2-dimensional space be given by its integer coordinates, $p = (p.x, p.y)$.

- A point p is said to be dominated by point q if $p.x \leq q.x$ and $p.y \leq q.y$.
- Given a set of n points, $P = \{p_1, p_2, \dots, p_n\}$ in 2-space a point is said to be maximal if it is not dominated by any other point in P .

The car selection problem can be modelled this way: For each car we associate (x, y) pair where x is the speed of the car and y is the negation of the price. High y value means a cheap car and low y means expensive car. Think of y as the money left in your pocket after you have paid for the car. Maximal points correspond to the fastest and cheapest cars.

The *2-dimensional Maxima* is thus defined as

- Given a set of points $P = \{p_1, p_2, \dots, p_n\}$ in 2-space, output the set of maximal points of P , i.e., those points p_i such that p_i is not dominated by any other point of P .

Here is set of maximal points for a given set of points in 2-d.

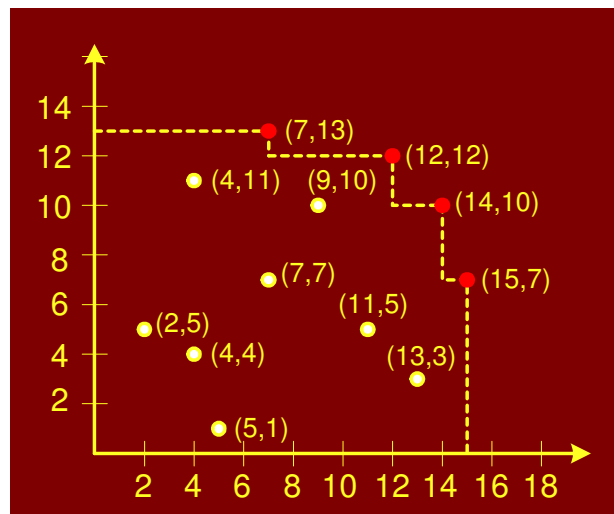


Figure 1.1: Maximal points in 2-d

Our description of the problem is at a fairly mathematical level. We have intentionally not discussed how the points are represented. We have not discussed any input or output formats. We have avoided programming and other software issues.

1.9 Brute-Force Algorithm

To get the ball rolling, let's just consider a simple brute-force algorithm, with no thought to efficiency. Let $P = \{p_1, p_2, \dots, p_n\}$ be the initial set of points. For each point p_i , test it against all other points p_j . If p_i is not dominated by any other point, then output it.

This English description is clear enough that any (competent) programmer should be able to implement it. However, if you want to be a bit more formal, it could be written in pseudocode as follows:

```

MAXIMA(int n, Point P[1...n])
1  for i ← 1 to n
2  do maximal ← true
3    for j ← 1 to n
4    do
5      if (i ≠ j) and (P[i].x ≤ P[j].x) and (P[i].y ≤ P[j].y)
6        then maximal ← false; break
7    if (maximal = true)
8      then output P[i]

```

There are no formal rules to the syntax of this pseudo code. In particular, do not assume that more detail is better. For example, I omitted type specifications for the procedure Maxima and the variable maximal, and I never defined what a Point data type is, since I felt that these are pretty clear from context or just unimportant details. Of course, the appropriate level of detail is a judgement call. Remember, algorithms are to be read by people, and so the level of detail depends on your intended audience. When writing pseudo code, you should omit details that detract from the main ideas of the algorithm, and just go with the essentials.

You might also notice that I did not insert any checking for consistency. For example, I assumed that the points in P are all distinct. If there is a duplicate point then the algorithm may fail to output even a single point. (Can you see why?) Again, these are important considerations for implementation, but we will often omit error checking because we want to see the algorithm in its simplest form.

Here are a series of figures that illustrate point domination.

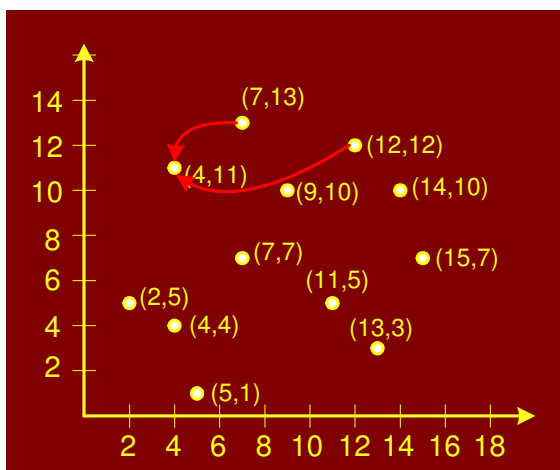


Figure 1.2: Points that dominate (4, 11)

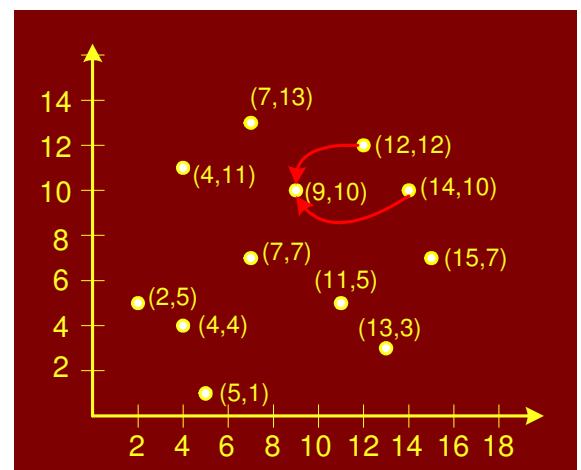


Figure 1.3: Points that dominate (9, 10)

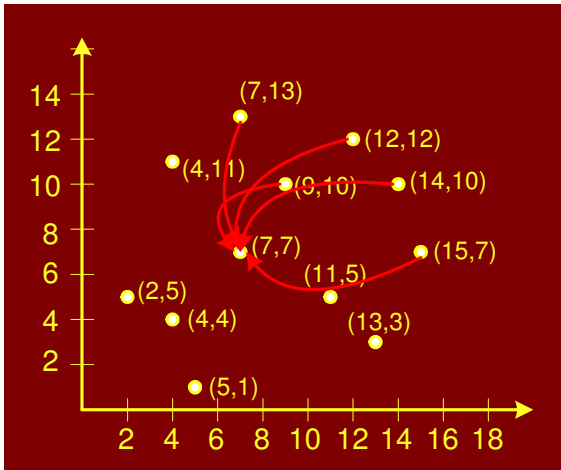


Figure 1.4: Points that dominate (7, 7)

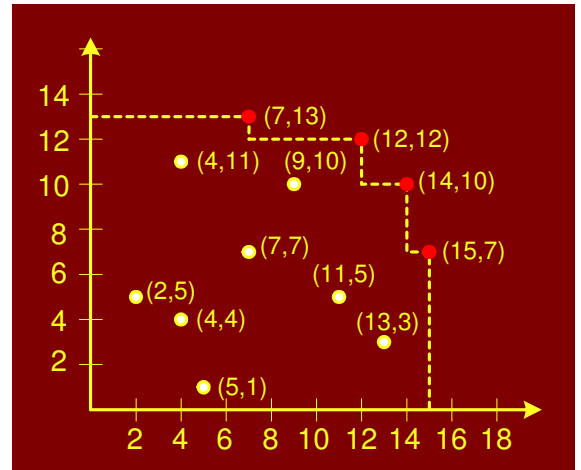


Figure 1.5: The maximal points

1.10 Running Time Analysis

The main purpose of our mathematical analysis will be measuring the execution time. We will also be concerned about the space (memory) required by the algorithm.

The running time of an implementation of the algorithm would depend upon the speed of the computer, programming language, optimization by the compiler etc. Although important, we will ignore these technological issues in our analysis.

To measure the running time of the brute-force 2-d maxima algorithm, we could count the number of steps of the pseudo code that are executed or, count the number of times an element of P is accessed or, the number of comparisons that are performed.

The running time depends upon the input size, e.g. n . Different inputs of the same size may result in different running time. For example, breaking out of the inner loop in the brute-force algorithm depends not only on the input size of P but also the structure of the input.

Two criteria for measuring running time are *worst-case time* and *average-case time*.

Worst-case time is the maximum running time over all (legal) inputs of size n . Let I denote an input instance, let $|I|$ denote its length, and let $T(I)$ denote the running time of the algorithm on input I . Then

$$T_{\text{worst}}(n) = \max_{|I|=n} T(I)$$

Average-case time is the average running time over all inputs of size n . Let $p(I)$ denote the probability of seeing this input. The average-case time is the weighted sum of running times with weights

being the probabilities:

$$T_{\text{avg}}(n) = \sum_{|I|=n} p(I)T(I)$$

We will almost always work with worst-case time. Average-case time is more difficult to compute; it is difficult to specify probability distribution on inputs. Worst-case time will specify an upper limit on the running time.

1.10.1 Analysis of the brute-force maxima algorithm.

Assume that the input size is n , and for the running time we will count the number of time that any element of P is accessed. Clearly we go through the outer loop n times, and for each time through this loop, we go through the inner loop n times as well. The condition in the if-statement makes four accesses to P . The output statement makes two accesses for each point that is output. In the worst case every point is maximal (can you see how to generate such an example?) so these two access are made for each time through the outer loop.

```

MAXIMA(int n, Point P[1...n])
1  for i ← 1 to n n times
2  do maximal ← true
3    for j ← 1 to n n times
4    do
5      if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y) 4 accesses
6        then maximal ← false break
7    if maximal
8      then output P[i].x, P[i].y 2 accesses

```

Thus we might express the worst-case running time as a pair of nested summations, one for the i -loop and the other for the j -loop:

$$\begin{aligned}
 T(n) &= \sum_{i=1}^n \left(2 + \sum_{j=1}^n 4 \right) \\
 &= \sum_{i=1}^n (4n + 2) \\
 &= (4n + 2)n = 4n^2 + 2n
 \end{aligned}$$

For small values of n , any algorithm is fast enough. What happens when n gets large? Running time does become an issue. When n is large, n^2 term will be much larger than the n term and will *dominate* the running time.

We will say that the worst-case running time is $\Theta(n^2)$. This is called the asymptotic growth rate of the function. We will discuss this Θ -notation more formally later.

The analysis involved computing a summation. Summation should be familiar but let us review a bit here. Given a finite sequence of values a_1, a_2, \dots, a_n , their sum $a_1 + a_2 + \dots + a_n$ is expressed in summation notation as

$$\sum_{i=1}^n a_i$$

If $n = 0$, then the sum is additive identity, 0.

Some facts about summation: If c is a constant

$$\sum_{i=1}^n c a_i = c \sum_{i=1}^n a_i$$

and

$$\sum_{i=1}^n (a_i + b_i) = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$$

Some important summations that should be committed to memory.

Arithmetic series

$$\begin{aligned} \sum_{i=1}^n i &= 1 + 2 + \dots + n \\ &= \frac{n(n+1)}{2} = \Theta(n^2) \end{aligned}$$

Quadratic series

$$\begin{aligned} \sum_{i=1}^n i^2 &= 1 + 4 + 9 + \dots + n^2 \\ &= \frac{2n^3 + 3n^2 + n}{6} = \Theta(n^3) \end{aligned}$$

Geometric series

$$\begin{aligned} \sum_{i=1}^n x^i &= 1 + x + x^2 + \dots + x^n \\ &= \frac{x^{(n+1)} - 1}{x - 1} = \Theta(n^2) \end{aligned}$$

If $0 < x < 1$ then this is $\Theta(1)$, and if $x > 1$, then this is $\Theta(x^n)$.

Harmonic series For $n \geq 0$

$$\begin{aligned} H_n &= \sum_{i=1}^n \frac{1}{i} \\ &= 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \approx \ln n \\ &= \Theta(\ln n) \end{aligned}$$

1.11 Analysis: A Harder Example

Let us consider a harder example.

```

NESTED-LOOPS()
1  for i ← 1 to n
2  do
3    for j ← 1 to 2i
4    do k = j ...
5      while (k ≥ 0)
6      do k = k - 1 ...

```

How do we analyze the running time of an algorithm that has complex nested loop? The answer is we write out the loops as summations and then solve the summations. To convert loops into summations, we work from inside-out.

Consider the *inner most while* loop.

```

NESTED-LOOPS()
1  for i ← 1 to n
2  do for j ← 1 to 2i
3    do k = j
4      while (k ≥ 0) ◀
5      do k = k - 1

```

It is executed for $k = j, j - 1, j - 2, \dots, 0$. Time spent inside the while loop is constant. Let $I()$ be the time spent in the while loop. Thus

$$I(j) = \sum_{k=0}^j 1 = j + 1$$

Consider the *middle for* loop.

```

NESTED-LOOPS()
1  for i ← 1 to n
2  do for j ← 1 to 2i ◀
3    do k = j
4      while (k ≥ 0)
5      do k = k - 1

```

Its running time is determined by i . Let $M(i)$ be the time spent in the for loop:

$$\begin{aligned}
 M(i) &= \sum_{j=1}^{2i} I(j) \\
 &= \sum_{j=1}^{2i} (j + 1) \\
 &= \sum_{j=1}^{2i} j + \sum_{j=1}^{2i} 1 \\
 &= \frac{2i(2i + 1)}{2} + 2i \\
 &= 2i^2 + 3i
 \end{aligned}$$

Finally the *outer-most* for loop.

```

NESTED-LOOPS()
1  for i ← 1 to n
2  do for j ← 1 to 2i
3    do k = j
4    while (k ≥ 0)
5    do k = k - 1
  
```

Let $T()$ be running time of the entire algorithm:

$$\begin{aligned}
 T(n) &= \sum_{i=1}^n M(i) \\
 &= \sum_{i=1}^n (2i^2 + 3i) \\
 &= \sum_{i=1}^n 2i^2 + \sum_{i=1}^n 3i \\
 &= 2 \frac{2n^3 + 3n^2 + n}{6} + 3 \frac{n(n+1)}{2} \\
 &= \frac{4n^3 + 15n^2 + 11n}{6} \\
 &= \Theta(n^3)
 \end{aligned}$$

1.11.1 2-dimension Maxima Revisited

Recall the 2-d maxima problem: Let a point p in 2-dimensional space be given by its integer coordinates, $p = (p.x, p.y)$. A point p is said to *dominated* by point q if $p.x \leq q.x$ and $p.y \leq q.y$. Given a set of n

points, $P = \{p_1, p_2, \dots, p_n\}$ in 2-space a point is said to be *maximal* if it is not dominated by any other point in P . The problem is to output all the maximal points of P . We introduced a brute-force algorithm that ran in $\Theta(n^2)$ time. It operated by comparing *all pairs* of points. Is there an approach that is significantly better?

The problem with the brute-force algorithm is that it uses no intelligence in *pruning* out decisions. For example, once we know that a point p_i is dominated by another point p_j , we do not need to use p_i for eliminating other points. This follows from the fact that dominance relation is *transitive*. If p_j dominates p_i and p_i dominates p_h then p_j also dominates p_h ; p_i is not needed.

1.11.2 Plane-sweep Algorithm

The question is whether we can make a significant improvement in the running time? Here is an idea for how we might do it. We will sweep a vertical line across the plane from left to right. As we sweep this line, we will build a structure holding the maximal points lying to the left of the sweep line. When the sweep line reaches the rightmost point of P , then we will have constructed the complete set of maxima. This approach of solving geometric problems by sweeping a line across the plane is called *plane sweep*.

Although we would like to think of this as a continuous process, we need some way to perform the plane sweep in discrete steps. To do this, we will begin by sorting the points in increasing order of their x -coordinates. For simplicity, let us assume that no two points have the same y -coordinate. (This limiting assumption is actually easy to overcome, but it is good to work with the simpler version, and save the messy details for the actual implementation.) Then we will advance the sweep-line from point to point in n discrete steps. As we encounter each new point, we will update the current list of maximal points.

We will sweep a vertical line across the 2-d plane from left to right. As we sweep, we will build a structure holding the maximal points lying to the left of the sweep line. When the sweep line reaches the rightmost point of P , we will have the complete set of maximal points. We will store the existing maximal points in a list. The points that p_i dominates will appear at the end of the list because points are sorted by x -coordinate. We will scan the list left to right. Every maximal point with y -coordinate less than p_i will be eliminated from computation. We will add maximal points onto the end of a list and delete from the end of the list. We can thus use a stack to store the maximal points. The point at the top of the stack will have the highest x -coordinate.

Here are a series of figures that illustrate the plane sweep. The figure also show the content of the stack.

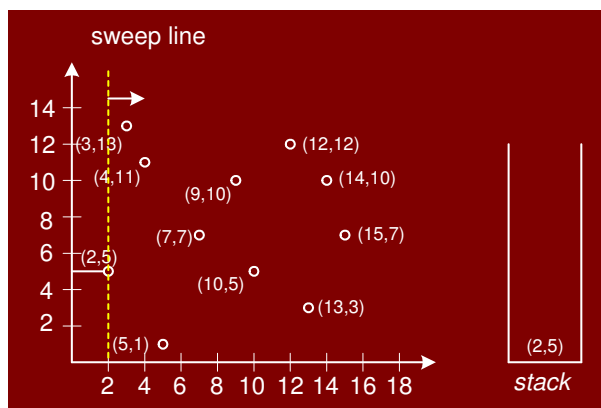


Figure 1.6: Sweep line at $(2, 5)$

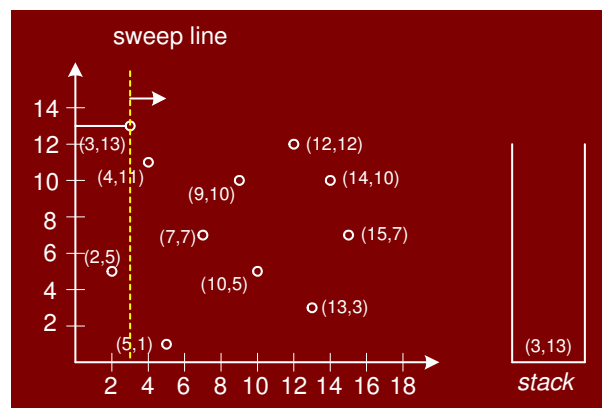


Figure 1.7: Sweep line at $(3, 13)$

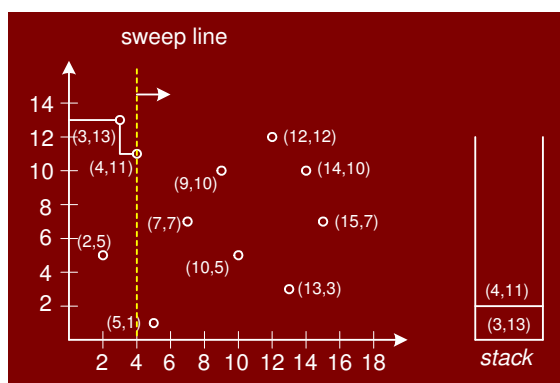


Figure 1.8: Sweep line at $(4, 11)$

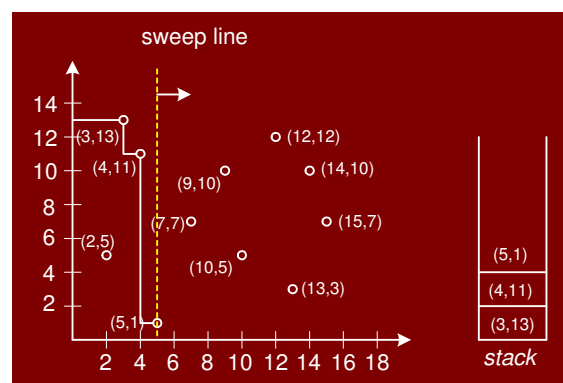


Figure 1.9: Sweep line at $(5, 1)$

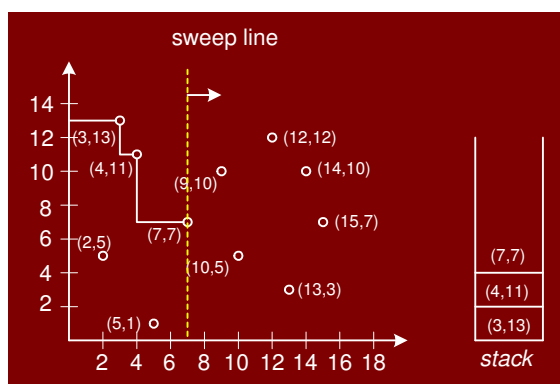


Figure 1.10: Sweep line at $(7, 7)$

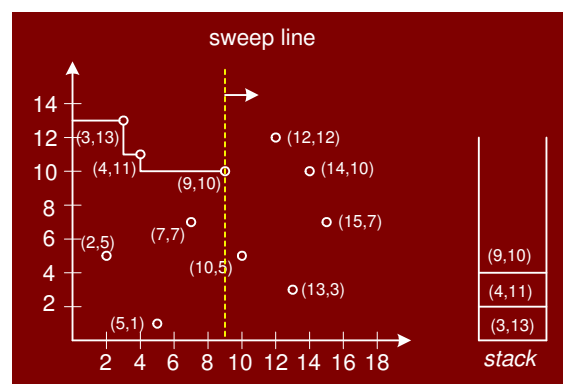


Figure 1.11: Sweep line at $(9, 10)$

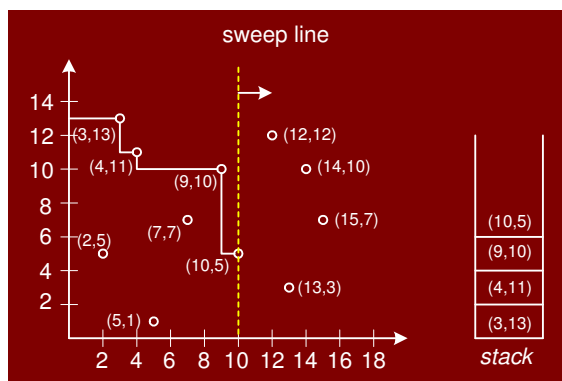


Figure 1.12: Sweep line at (10, 5)

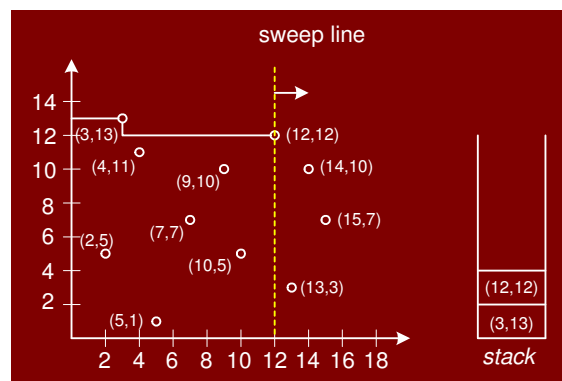


Figure 1.13: Sweep line at (12, 12)

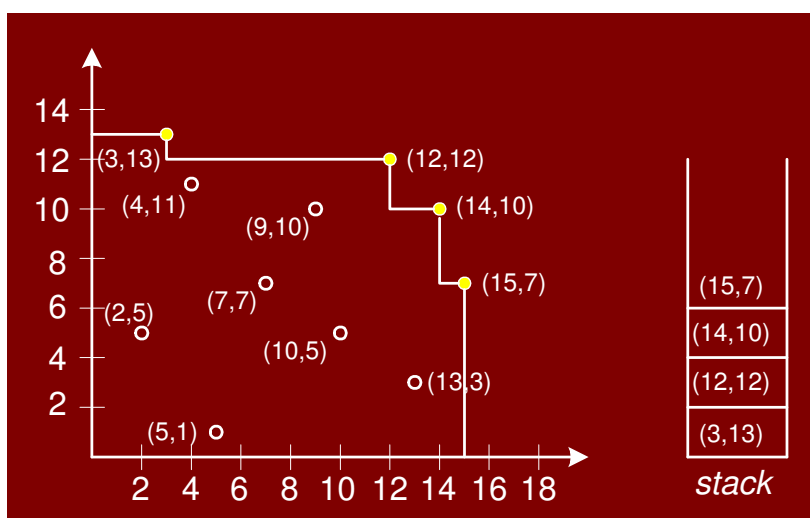


Figure 1.14: The final maximal set

Here is the algorithm.

PLANE-SWEEP-MAXIMA(n , $P[1..n]$)

- 1 sort P in increasing order by x ;
- 2 stack s ;
- 3 **for** $i \leftarrow 1$ **to** n
- 4 **do**
- 5 **while** (s .notEmpty() & s .top(). $y \leq P[i].y$)
- 6 **do** s .pop();
- 7 s .push($P[i]$);
- 8 output the contents of stack s ;

1.11.3 Analysis of Plane-sweep Algorithm

Sorting takes $\Theta(n \log n)$; we will show this later when we discuss sorting. The for loop executes n times. The inner loop (seemingly) could be iterated $(n - 1)$ times. It seems we still have an $n(n - 1)$ or $\Theta(n^2)$ algorithm. Got fooled by simple minded loop-counting. The while loop will not execute more than n times over the entire course of the algorithm. Why is this? Observe that the total number of elements that can be pushed on the stack is n since we execute exactly one push each time during the outer for-loop.

We pop an element off the stack each time we go through the inner while-loop. It is impossible to pop more elements than are ever pushed on the stack. Therefore, the inner while-loop cannot execute more than n times over the entire course of the algorithm. (*Make sure that you understand this*).

The for-loop iterates n times and the inner while-loop also iterates n time for a total of $\Theta(n)$. Combined with the sorting, the runtime of entire plane-sweep algorithm is $\Theta(n \log n)$.

1.11.4 Comparison of Brute-force and Plane sweep algorithms

How much of an improvement is plane-sweep over brute-force? Consider the ratio of running times:

$$\frac{n^2}{n \log n} = \frac{n}{\log n}$$

n	$\log n$	$\frac{n}{\log n}$
100	7	15
1000	10	100
10000	13	752
100000	17	6021
1000000	20	50171

For $n = 1,000,000$, if plane-sweep takes 1 second, the brute-force will take about 14 hours!. From this we get an idea about the importance of asymptotic analysis. It tells us which algorithm is better for large values of n . As we mentioned before, if n is not very large, then almost any algorithm will be fast. But efficient algorithm design is most important for large inputs, and the general rule of computing is that input sizes continue to grow until people can no longer tolerate the running times. Thus, by designing algorithms efficiently, you make it possible for the user to run large inputs in a reasonable amount of time.

Chapter 2

Asymptotic Notation

You may be asking that we continue to use the notation $\Theta()$ but have never defined it. Let's remedy this now. Given any function $g(n)$, we define $\Theta(g(n))$ to be a set of functions that *asymptotically equivalent* to $g(n)$. Formally:

$$\Theta(g(n)) = \{f(n) \mid \text{there exist positive constants } c_1, c_2 \text{ and } n_0 \text{ such that } 0 \leq c_1g(n) \leq f(n) \leq c_2g(n) \text{ for all } n \geq n_0\}$$

This is written as " $f(n) \in \Theta(g(n))$ " That is, $f(n)$ and $g(n)$ are *asymptotically equivalent*. This means that they have essentially the same growth rates for large n . For example, functions like

- $4n^2$,
- $(8n^2 + 2n - 3)$,
- $(n^2/5 + \sqrt{n} - 10 \log n)$
- $n(n - 3)$

are all asymptotically equivalent. As n becomes large, the *dominant* (fastest growing) term is some constant times n^2 .

Consider the function

$$f(n) = 8n^2 + 2n - 3$$

Our informal rule of keeping the largest term and ignoring the constant suggests that $f(n) \in \Theta(n^2)$. Let's see why this bears out formally. We need to show two things for

$$f(n) = 8n^2 + 2n - 3$$

Lower bound $f(n) = 8n^2 + 2n - 3$ grows asymptotically at least as fast as n^2 ,

Upper bound $f(n)$ grows no faster asymptotically than n^2 ,

Lower bound: $f(n)$ grows asymptotically at least as fast as n^2 . For this, need to show that there exist positive constants c_1 and n_0 , such that $f(n) \geq c_1 n^2$ for all $n \geq n_0$. Consider the reasoning

$$f(n) = 8n^2 + 2n - 3 \geq 8n^2 - 3 = 7n^2 + (n^2 - 3) \geq 7n^2$$

Thus $c_1 = 7$. We implicitly assumed that $2n \geq 0$ and $n^2 - 3 \geq 0$. These are not true for all n but if $n \geq \sqrt{3}$, then both are true. So select $n_0 \geq \sqrt{3}$. We then have $f(n) \geq c_1 n^2$ for all $n \geq n_0$.

Upper bound: $f(n)$ grows asymptotically no faster than n^2 . For this, we need to show that there exist positive constants c_2 and n_0 , such that $f(n) \leq c_2 n^2$ for all $n \geq n_0$. Consider the reasoning

$$f(n) = 8n^2 + 2n - 3 \leq 8n^2 + 2n \leq 8n^2 + 2n^2 = 10n^2$$

Thus $c_2 = 10$. We implicitly made the assumption that $2n \leq 2n^2$. This is not true for all n but it is true for all $n \geq 1$. So select $n_0 \geq 1$. We thus have $f(n) \leq c_2 n^2$ for all $n \geq n_0$.

From lower bound we have $n_0 \geq \sqrt{3}$. From upper bound we have $n_0 \geq 1$. Combining the two, we let n_0 be the larger of the two: $n_0 \geq \sqrt{3}$. In conclusion, if we let $c_1 = 7$, $c_2 = 10$ and $n_0 \geq \sqrt{3}$, we have

$$7n^2 \leq 8n^2 + 2n - 3 \leq 10n^2 \quad \text{for all } n \geq \sqrt{3}$$

We have thus established

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \quad \text{for all } n \geq n_0$$

Here are plots of the three functions. Notice the bounds.

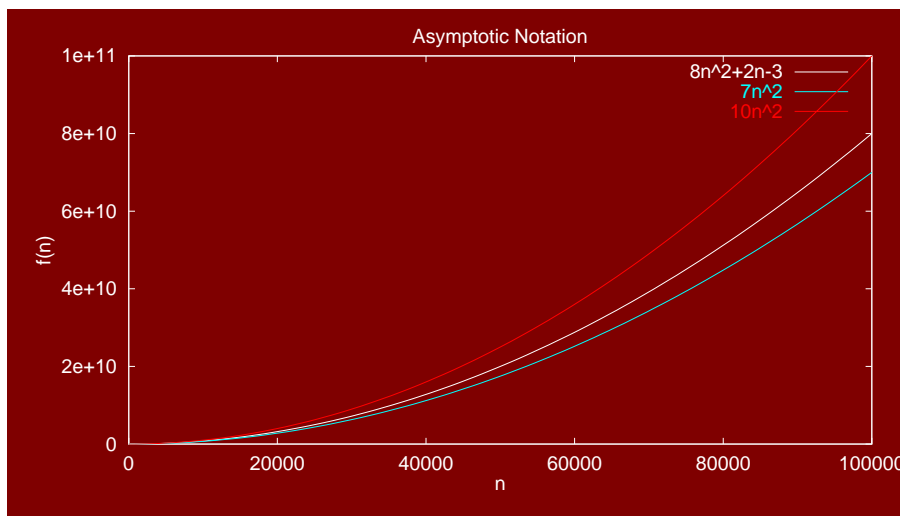


Figure 2.1: Asymptotic Notation Example

We have established that $f(n) \in n^2$. Let's show why $f(n)$ is not in some other asymptotic class. First, let's show that $f(n) \notin \Theta(n)$. Show that $f(n) \notin \Theta(n)$. If this were true, we would have had to satisfy

both the upper and lower bounds. The lower bound is satisfied because $f(n) = 8n^2 + 2n - 3$ does grow at least as fast asymptotically as n . But the upper bound is false. Upper bounds requires that there exist positive constants c_2 and n_0 such that $f(n) \leq c_2n$ for all $n \geq n_0$.

Informally we know that $f(n) = 8n^2 + 2n - 3$ will eventually exceed c_2n no matter how large we make c_2 . To see this, suppose we assume that constants c_2 and n_0 did exist such that $8n^2 + 2n - 3 \leq c_2n$ for all $n \geq n_0$. Since this is true for all sufficiently large n then it must be true in the limit as n tends to infinity. If we divide both sides by n , we have

$$\lim_{n \rightarrow \infty} \left(8n + 2 - \frac{3}{n} \right) \leq c_2.$$

It is easy to see that in the limit, the left side tends to ∞ . So, no matter how large c_2 is, the statement is violated. Thus $f(n) \notin \Theta(n)$.

Let's show that $f(n) \notin \Theta(n^3)$. The idea would be to show that the lower bound $f(n) \geq c_1n^3$ for all $n \geq n_0$ is violated. (c_1 and n_0 are positive constants). Informally we know this to be true because any cubic function will overtake a quadratic.

If we divide both sides by n^3 :

$$\lim_{n \rightarrow \infty} \left(\frac{8}{n} + \frac{2}{n^2} - \frac{3}{n^3} \right) \geq c_1$$

The left side tends to 0. The only way to satisfy this is to set $c_1 = 0$. But by hypothesis, c_1 is positive. This means that $f(n) \notin \Theta(n^3)$.

The definition of Θ -notation relies on proving both a lower and upper asymptotic bound. Sometimes we are only interested in proving one bound or the other. The O -notation is used to state only the asymptotic *upper* bounds.

$$O(g(n)) = \{f(n) \mid \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0\}$$

The Ω -notation allows us to state only the asymptotic *lower* bounds.

$$\Omega(g(n)) = \{f(n) \mid \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\}$$

The three notations:

$$\Theta(g(n)) : 0 \leq c_1g(n) \leq f(n) \leq c_2g(n)$$

$$O(g(n)) : 0 \leq f(n) \leq cg(n)$$

$$\Omega(g(n)) : 0 \leq cg(n) \leq f(n)$$

for all $n \geq n_0$

These definitions suggest an alternative way of showing the asymptotic behavior. We can use limits for define the asymptotic behavior. Limit rule for Θ -notation:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c,$$

for some constant $c > 0$ (strictly positive but not infinity) then $f(n) \in \Theta(g(n))$. Similarly, the limit rule for O -notation is

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c,$$

for some constant $c \geq 0$ (nonnegative but not infinite) then $f(n) \in O(g(n))$ and limit rule for Ω -notation:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq 0,$$

(either a strictly positive constant or infinity) then $f(n) \in \Omega(g(n))$.

Here is a list of common asymptotic running times:

- $\Theta(1)$: Constant time; can't beat it!
- $\Theta(\log n)$: Inserting into a balanced binary tree; time to find an item in a sorted array of length n using binary search.
- $\Theta(n)$: About the fastest that an algorithm can run.
- $\Theta(n \log n)$: Best sorting algorithms.
- $\Theta(n^2)$, $\Theta(n^3)$: Polynomial time. These running times are acceptable when the exponent of n is small or n is not too large, e.g., $n \leq 1000$.
- $\Theta(2^n)$, $\Theta(3^n)$: Exponential time. Acceptable only if n is small, e.g., $n \leq 50$.
- $\Theta(n!)$, $\Theta(n^n)$: Acceptable only for really small n , e.g. $n \leq 20$.

Chapter 3

Divide and Conquer Strategy

The ancient Roman politicians understood an important principle of good algorithm design (although they were probably not thinking about algorithms at the time). You divide your enemies (by getting them to distrust each other) and then conquer them piece by piece. This is called *divide-and-conquer*. In algorithm design, the idea is to take a problem on a large input, break the input into smaller pieces, solve the problem on each of the small pieces, and then combine the piecewise solutions into a global solution. But once you have broken the problem into pieces, how do you solve these pieces? The answer is to apply divide-and-conquer to them, thus further breaking them down. The process ends when you are left with such tiny pieces remaining (e.g. one or two items) that it is trivial to solve them. Summarizing, the main elements to a divide-and-conquer solution are

Divide: the problem into a small number of pieces

Conquer: solve each piece by applying divide and conquer to it *recursively*

Combine: the pieces together into a global solution.

3.1 Merge Sort

Divide and conquer strategy is applicable in a huge number of computational problems. The first example of divide and conquer algorithm we will discuss is a simple and efficient sorting procedure called Merge Sort. We are given a sequence of n numbers A , which we will assume are stored in an array $A[1..n]$. The objective is to output a permutation of this sequence sorted in increasing order. This is normally done by permuting the elements within the array A . Here is how the merge sort algorithm works:

- (Divide:) split A down the middle into two subsequences, each of size roughly $n/2$
- (Conquer:) sort each subsequence by calling merge sort recursively on each.
- (Combine:) merge the two sorted subsequences into a single sorted list.

Let's design the algorithm top-down. We'll assume that the procedure that merges two sorted lists is available to us. We'll implement it later. Because the algorithm is called recursively on sublists, in addition to passing in the array itself, we will pass in two indices, which indicate the first and last indices of the sub-array that we are to sort. The call $\text{MergeSort}(A, p, r)$ will sort the sub-array $A[p : r]$ and return the sorted result in the same sub-array. Here is the overview. If $r = p$, then this means that there is only one element to sort, and we may return immediately. Otherwise if $(p \neq r)$ there are at least two elements, and we will invoke the divide-and-conquer. We find the index q , midway between p and r , namely $q = (p + r)/2$ (rounded down to the nearest integer). Then we split the array into sub-arrays $A[p : q]$ and $A[q + 1 : r]$. Call MergeSort recursively to sort each sub-array. Finally, we invoke a procedure (which we have yet to write) which merges these two sub-arrays into a single sorted array.

Here is the MergeSort algorithm.

```

MERGE-SORT( array A, int p, int r)
1  if (p < r)
2    then
3       $q \leftarrow (p + r)/2$ 
4      MERGE-SORT( $A, p, q$ ) // sort  $A[p..q]$ 
5      MERGE-SORT( $A, q + 1, r$ ) // sort  $A[q + 1..r]$ 
6      MERGE( $A, p, q, r$ ) // merge the two pieces

```

The following figure illustrates the dividing (splitting) procedure.

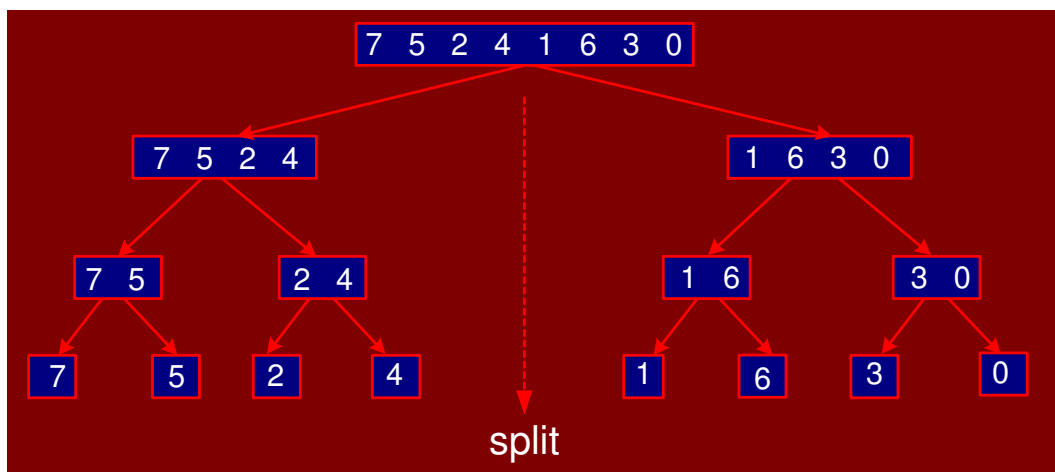


Figure 3.1: Merge sort divide phase

Merging: All that is left is to describe the procedure that merges two sorted lists. $\text{Merge}(A, p, q, r)$ assumes that the left sub-array, $A[p : q]$, and the right sub-array, $A[q + 1 : r]$, have already been sorted. We merge these two sub-arrays by copying the elements to a temporary working array called B . For convenience, we will assume that the array B has the same index range A , that is, $B[p : r]$. (One nice thing about pseudo-code, is that we can make these assumptions, and leave them up to the programmer to figure out how to implement it.) We have to indices i and j , that point to the current elements of each

sub-array. We move the smaller element into the next position of B (indicated by index k) and then increment the corresponding index (either i or j). When we run out of elements in one array, then we just copy the rest of the other array into B. Finally, we copy the entire contents of B back into A. (The use of the temporary array is a bit unpleasant, but this is impossible to overcome entirely. It is one of the shortcomings of MergeSort, compared to some of the other efficient sorting algorithms.)

Here is the merge algorithm:

```

MERGE( array A, int p, int q, int r)
1  int B[p..r]; int i ← k ← p; int j ← q + 1
2  while (i ≤ q) and (j ≤ r)
3  do if (A[i] ≤ A[j])
4      then B[k++] ← A[i++]
5      else B[k++] ← A[j++]
6  while (i ≤ q)
7  do B[k++] ← A[i++]
8  while (j ≤ r)
9  do B[k++] ← A[j++]
10 for i ← p to r
11 do A[i] ← B[i]

```

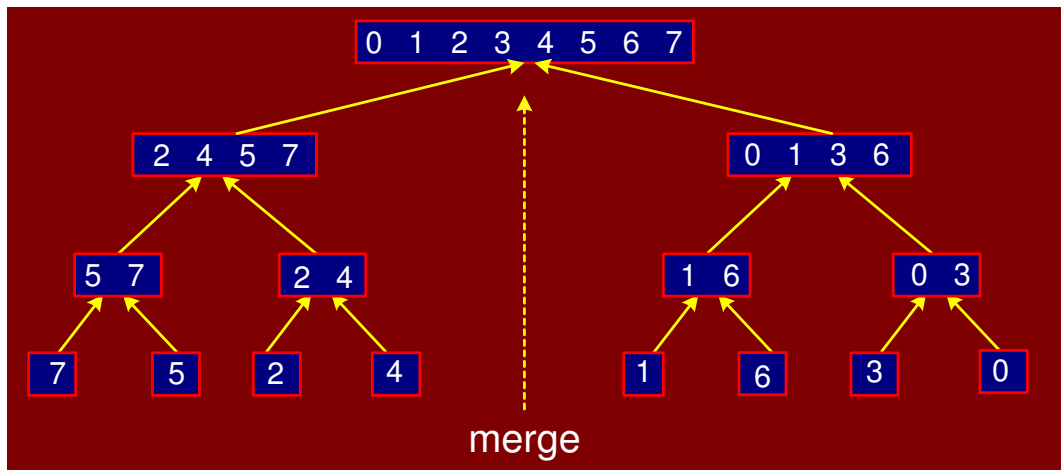


Figure 3.2: Merge sort: combine phase

3.1.1 Analysis of Merge Sort

First let us consider the running time of the procedure $\text{Merge}(A, p, q, r)$. Let $n = r - p + 1$ denote the total length of both the left and right sub-arrays. What is the running time of Merge as a function of n ? The algorithm contains four loops (none nested in the other). It is easy to see that each loop can be executed at most n times. (If you are a bit more careful you can actually see that all the while-loops

together can only be executed n times in total, because each execution copies one new element to the array B , and B only has space for n elements.) Thus the running time to Merge n items is $\Theta(n)$. Let us write this without the asymptotic notation, simply as n . (We'll see later why we do this.)

Now, how do we describe the running time of the entire MergeSort algorithm? We will do this through the use of a recurrence, that is, a function that is defined recursively in terms of itself. To avoid circularity, the recurrence for a given value of n is defined in terms of values that are strictly smaller than n . Finally, a recurrence has some basis values (e.g. for $n = 1$), which are defined explicitly.

Let $T(n)$ denote the worst case running time of MergeSort on an array of length n . If we call MergeSort with an array containing a single item ($n = 1$) then the running time is constant. We can just write $T(n) = 1$, ignoring all constants. For $n > 1$, MergeSort splits into two halves, sorts the two and then merges them together. The left half is of size $\lceil n/2 \rceil$ and the right half is $\lfloor n/2 \rfloor$. How long does it take to sort elements in sub array of size $\lceil n/2 \rceil$? We do not know this but because $\lceil n/2 \rceil < n$ for $n > 1$, we can express this as $T(\lceil n/2 \rceil)$. Similarly the time taken to sort right sub array is expressed as $T(\lfloor n/2 \rfloor)$. In conclusion we have

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n & \text{otherwise} \end{cases}$$

This is called *recurrence relation*, i.e., a recursively defined function. Divide-and-conquer is an important design technique, and it naturally gives rise to recursive algorithms. It is thus important to develop mathematical techniques for solving recurrences, either exactly or asymptotically.

Let's expand the terms.

$$\begin{aligned} T(1) &= 1 \\ T(2) &= T(1) + T(1) + 2 = 1 + 1 + 2 = 4 \\ T(3) &= T(2) + T(1) + 3 = 4 + 1 + 3 = 8 \\ T(4) &= T(2) + T(2) + 4 = 4 + 4 + 4 = 12 \\ T(5) &= T(3) + T(2) + 5 = 8 + 4 + 5 = 17 \\ &\dots \\ T(8) &= T(4) + T(4) + 8 = 12 + 12 + 8 = 32 \\ &\dots \\ T(16) &= T(8) + T(8) + 16 = 32 + 32 + 16 = 80 \\ &\dots \\ T(32) &= T(16) + T(16) + 32 = 80 + 80 + 32 = 192 \end{aligned}$$

What is the pattern here? Let's consider the ratios $T(n)/n$ for powers of 2:

$$\begin{aligned} T(1)/1 &= 1 & T(8)/8 &= 4 \\ T(2)/2 &= 2 & T(16)/16 &= 5 \\ T(4)/4 &= 3 & T(32)/32 &= 6 \end{aligned}$$

This suggests $T(n)/n = \log n + 1$ Or, $T(n) = n \log n + n$ which is $\Theta(n \log n)$ (using the limit rule).

3.1.2 The Iteration Method for Solving Recurrence Relations

Floor and ceilings are a pain to deal with. If n is assumed to be a power of 2 ($2^k = n$), this will simplify the recurrence to

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2T(n/2) + n & \text{otherwise} \end{cases}$$

The iteration method turns the recurrence into a summation. Let's see how it works. Let's expand the recurrence:

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2(2T(n/4) + n/2) + n \\ &= 4T(n/4) + n + n \\ &= 4(2T(n/8) + n/4) + n + n \\ &= 8T(n/8) + n + n + n \\ &= 8(2T(n/16) + n/8) + n + n + n \\ &= 16T(n/16) + n + n + n + n \end{aligned}$$

If n is a power of 2 then let $n = 2^k$ or $k = \log n$.

$$\begin{aligned} T(n) &= 2^k T(n/(2^k)) + \underbrace{(n + n + n + \dots + n)}_{k \text{ times}} \\ &= 2^k T(n/(2^k)) + kn \\ &= 2^{(\log n)} T(n/(2^{(\log n)})) + (\log n)n \\ &= 2^{(\log n)} T(n/n) + (\log n)n \\ &= nT(1) + n \log n = n + n \log n \end{aligned}$$

3.1.3 Visualizing Recurrences Using the Recursion Tree

Iteration is a very powerful technique for solving recurrences. But, it is easy to get lost in all the symbolic manipulations and lose sight of what is going on. Here is a nice way to visualize what is going on in iteration. We can describe any recurrence in terms of a tree, where each expansion of the recurrence takes us one level deeper in the tree.

Recall that the recurrence for MergeSort (which we simplified by assuming that n is a power of 2, and hence could drop the floors and ceilings)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2T(n/2) + n & \text{otherwise} \end{cases}$$

Suppose that we draw the recursion tree for MergeSort, but each time we merge two lists, we label that node of the tree with the time it takes to perform the associated (nonrecursive) merge. Recall that to

merge two lists of size $m/2$ to a list of size m takes $\Theta(m)$ time, which we will just write as m . Below is an illustration of the resulting recursion tree.

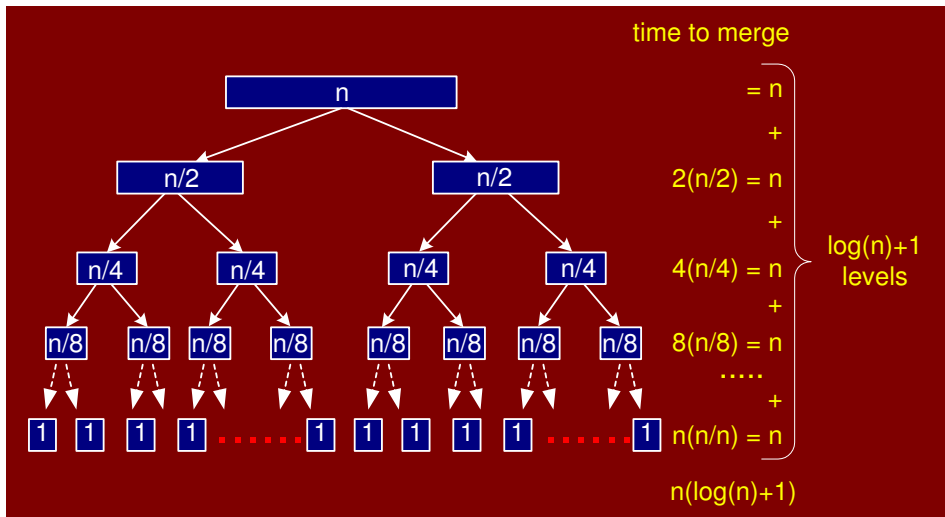


Figure 3.3: Merge sort Recurrence Tree

3.1.4 A Messier Example

The merge sort recurrence was too easy. Let's try a messier recurrence.

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 3T(n/4) + n & \text{otherwise} \end{cases}$$

Assume n to be a power of 4, i.e., $n = 4^k$ and $k = \log_4 n$

$$\begin{aligned} T(n) &= 3T(n/4) + n \\ &= 3(3T(n/16) + (n/4)) + n \\ &= 9T(n/16) + 3(n/4) + n \\ &= 27T(n/64) + 9(n/16) + 3(n/4) + n \\ &= \dots \\ &= 3^k T\left(\frac{n}{4^k}\right) + 3^{k-1}\left(\frac{n}{4^{k-1}}\right) \\ &\quad + \dots + 9(n/16) + 3(n/4) + n \\ &= 3^k T\left(\frac{n}{4^k}\right) + \sum_{i=0}^{k-1} \frac{3^i}{4^i} n \end{aligned}$$

With $n = 4^k$ and $T(1) = 1$

$$\begin{aligned} T(n) &= 3^k T\left(\frac{n}{4^k}\right) + \sum_{i=0}^{k-1} \frac{3^i}{4^i} n \\ &= 3^{\log_4 n} T(1) + \sum_{i=0}^{(\log_4 n)-1} \frac{3^i}{4^i} n \\ &= n^{\log_4 3} + \sum_{i=0}^{(\log_4 n)-1} \frac{3^i}{4^i} n \end{aligned}$$

We used the formula $a^{\log_b n} = n^{\log_b a}$. n remains constant throughout the sum and $3^i/4^i = (3/4)^i$; we thus have

$$T(n) = n^{\log_4 3} + n \sum_{i=0}^{(\log_4 n)-1} \left(\frac{3}{4}\right)^i$$

The sum is a geometric series; recall that for $x \neq 1$

$$\sum_{i=0}^m x^i = \frac{x^{m+1} - 1}{x - 1}$$

In this case $x = 3/4$ and $m = \log_4 n - 1$. We get

$$T(n) = n^{\log_4 3} + n \frac{(3/4)^{\log_4 n + 1} - 1}{(3/4) - 1}$$

Applying the log identity once more

$$\begin{aligned} (3/4)^{\log_4 n} &= n^{\log_4 (3/4)} = n^{\log_4 3 - \log_4 4} \\ &= n^{\log_4 3 - 1} = \frac{n^{\log_4 3}}{n} \end{aligned}$$

If we plug this back, we get

$$\begin{aligned} T(n) &= n^{\log_4 3} + n \frac{\frac{n^{\log_4 3}}{n} - 1}{(3/4) - 1} \\ &= n^{\log_4 3} + \frac{n^{\log_4 3} - n}{-1/4} \\ &= n^{\log_4 3} + 4(n - n^{\log_4 3}) \\ &= 4n - 3n^{\log_4 3} \end{aligned}$$

With $\log_4 3 \approx 0.79$, we finally have the result!

$$T(n) = 4n - 3n^{\log_4 3} \approx 4n - 3n^{0.79} \in \Theta(n)$$

3.2 Selection Problem

Suppose we are given a set of n numbers. Define the *rank* of an element to be one plus the number of elements that are smaller. Thus, the rank of an element is its final position if the set is sorted. The minimum is of rank 1 and the maximum is of rank n .

Consider the set: $\{5, 7, 2, 10, 8, 15, 21, 37, 41\}$. The rank of each number is its position in the sorted order.

<i>position</i>	1	2	3	4	5	6	7	8	9
Number	2	5	7	8	10	15	21	37	41

For example, rank of 8 is 4, one plus the number of elements smaller than 8 which is 3.

The selection problem is stated as follows:

Given a set A of n distinct numbers and an integer k , $1 \leq k \leq n$, output the element of A of rank k .

Of particular interest in statistics is the *median*. If n is odd then the median is defined to be element of rank $(n + 1)/2$. When n is even, there are two choices: $n/2$ and $(n + 1)/2$. In statistics, it is common to return the average of the two elements.

Medians are useful as a measures of the *central tendency* of a set especially when the distribution of values is highly skewed. For example, the median income in a community is a more meaningful measure than average. Suppose 7 households have monthly incomes 5000, 7000, 2000, 10000, 8000, 15000 and 16000. In sorted order, the incomes are 2000, 5000, 7000, 8000, 10000, 15000, 16000. The median income is 8000; median is element with rank 4: $(7 + 1)/2 = 4$. The average income is 9000. Suppose the income 16000 goes up to 450,000. The median is still 8000 but the average goes up to 71,000. Clearly, the average is not a good representative of the majority income levels.

The selection problem can be easily solved by simply sorting the numbers of A and returning $A[k]$. Sorting, however, requires $\Theta(n \log n)$ time. The question is: can we do better than that? In particular, is it possible to solve the selections problem in $\Theta(n)$ time? The answer is yes. However, the solution is far from obvious.

3.2.1 Sieve Technique

The reason for introducing this algorithm is that it illustrates a very important special case of divide-and-conquer, which I call the *sieve technique*. We think of divide-and-conquer as breaking the problem into a small number of smaller subproblems, which are then solved recursively. The sieve technique is a special case, where the number of subproblems is just 1.

The sieve technique works in phases as follows. It applies to problems where we are interested in finding a single item from a larger set of n items. We do not know which item is of interest, however after doing some amount of analysis of the data, taking say $\Theta(nk)$ time, for some constant k , we find that we do not

know what the desired item is, but we can identify a large enough number of elements that cannot be the desired value, and can be eliminated from further consideration. In particular “large enough” means that the number of items is at least some fixed constant fraction of n (e.g. $n/2$, $n/3$). Then we solve the problem recursively on whatever items remain. Each of the resulting recursive solutions then do the same thing, eliminating a constant fraction of the remaining set.

3.2.2 Applying the Sieve to Selection

To see more concretely how the sieve technique works, let us apply it to the selection problem. We will begin with the given array $A[1..n]$. We will pick an item from the array, called the *pivot element* which we will denote by x . We will talk about how an item is chosen as the pivot later; for now just think of it as a random element of A .

We then *partition* A into three parts.

1. $A[q]$ contains the pivot element x ,
2. $A[1..q - 1]$ will contain all the elements that are less than x and
3. $A[q + 1..n]$ will contain all elements that are greater than x .

Within each sub array, the items may appear in any order. The following figure shows a partitioned array:

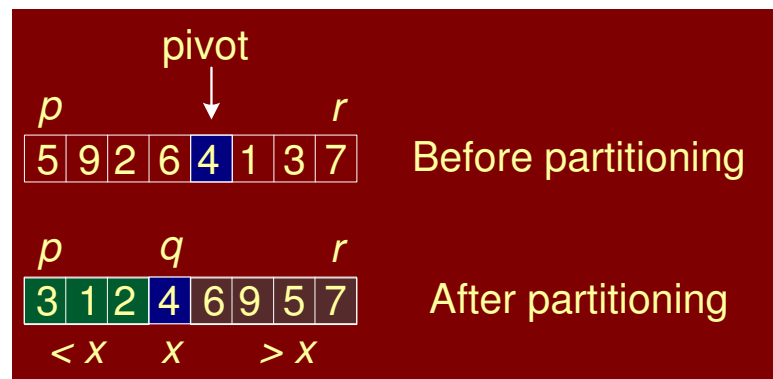


Figure 3.4: $A[p..r]$ partitioned about the pivot x

3.2.3 Selection Algorithm

It is easy to see that the rank of the pivot x is $q - p + 1$ in $A[p..r]$. Let $\text{rank}_x = q - p + 1$. If $k = \text{rank}_x$ then the pivot is k^{th} smallest. If $k < \text{rank}_x$ then search $A[p..q - 1]$ recursively. If $k > \text{rank}_x$ then search $A[q + 1..r]$ recursively. Find element of rank $(k - q)$ because we eliminated q smaller elements in A .

SELECT(array A , int p , int r , int k)

```

1  if (p = r)
2    then return A[p]
3  else x ← CHOOSE_PIVOT(A, p, r)
4      q ← PARTITION(A, p, r, x)
5      rank_x ← q - p + 1
6      if k = rank_x
7        then return x
8      if k < rank_x
9        then return SELECT(A, p, q - 1, k)
10     else return SELECT(A, q + 1, r, k - q)

```

Example: select the 6th smallest element of the set {5, 9, 2, 6, 4, 1, 3, 7}

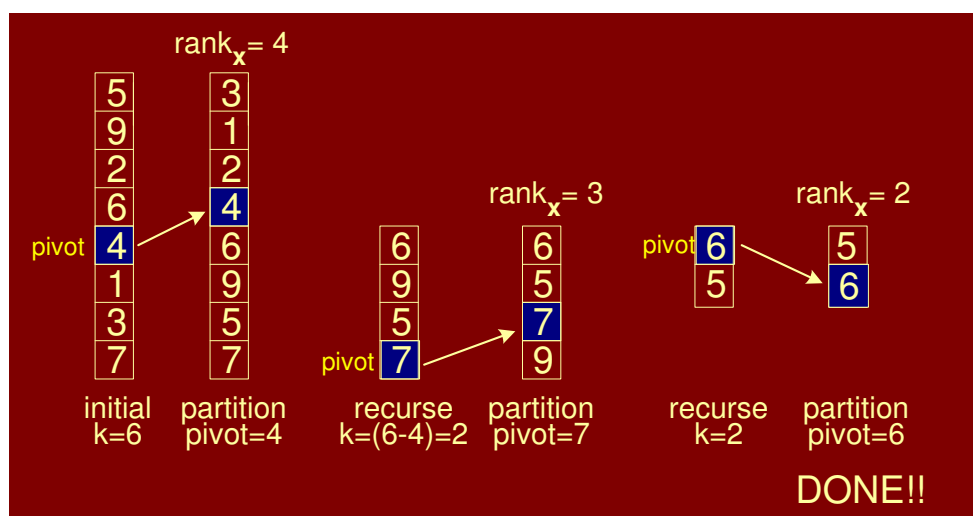


Figure 3.5: Sieve example: select 6th smallest element

3.2.4 Analysis of Selection

We will discuss how to choose a pivot and the partitioning later. For the moment, we will assume that they both take $\Theta(n)$ time. How many elements do we eliminate in each time? If x is the largest or the smallest then we may only succeed in eliminating one element.

5, 9, 2, 6, 4, 1, 3, 7 pivot is 1
1, 5, 9, 2, 6, 4, 3, 7 after partition

Ideally, x should have a rank that is neither too large or too small.

Suppose we are able to choose a pivot that causes exactly half of the array to be eliminated in each phase.

This means that we recurse on the remaining $n/2$ elements. This leads to the following recurrence:

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(n/2) + n & \text{otherwise} \end{cases}$$

If we expand this recurrence, we get

$$\begin{aligned} T(n) &= n + \frac{n}{2} + \frac{n}{4} + \dots \\ &\leq \sum_{i=0}^{\infty} \frac{n}{2^i} \\ &= n \sum_{i=0}^{\infty} \frac{1}{2^i} \end{aligned}$$

Recall the formula for infinite geometric series; for any $|c| < 1$,

$$\sum_{i=0}^{\infty} c^i = \frac{1}{1-c}$$

Using this we have

$$T(n) \leq 2n \in \Theta(n)$$

Let's think about how we ended up with a $\Theta(n)$ algorithm for selection. Normally, a $\Theta(n)$ time algorithm would make a single or perhaps a constant number of passes of the data set. In this algorithm, we make a number of passes. In fact it could be as many as $\log n$.

However, because we eliminate a constant fraction of the array with each phase, we get the convergent geometric series in the analysis. This shows that the total running time is indeed *linear* in n . This lesson is well worth remembering. It is often possible to achieve linear running times in ways that you would not expect.

Chapter 4

Sorting

For the next series of lectures, we will focus on sorting. There are a number of reasons for sorting. Here are a few important ones. Procedures for sorting are parts of many large software systems. Design of efficient sorting algorithms is necessary to achieve overall efficiency of these systems.

Sorting is a well studied problem from the analysis point of view. Sorting is one of the few problems where provable lower bounds exist on how fast we can sort. In sorting, we are given an array $A[1..n]$ of n numbers. We are to reorder these elements into increasing (or decreasing) order.

More generally, A is an array of objects and we sort them based on one of the attributes - the *key value*. The key value need not be a number. It can be any object from a *totally ordered domain*. Totally ordered domain means that for any two elements of the domain, x and y , either $x < y$, $x = y$ or $x > y$.

4.1 Slow Sorting Algorithms

There are a number of well-known slow $O(n^2)$ sorting algorithms. These include the following:

Bubble sort: Scan the array. Whenever two consecutive items are found that are out of order, swap them. Repeat until all consecutive items are in order.

Insertion sort: Assume that $A[1..i - 1]$ have already been sorted. Insert $A[i]$ into its proper position in this sub array. Create this position by shifting all larger elements to the right.

Selection sort: Assume that $A[1..i - 1]$ contain the $i - 1$ smallest elements in sorted order. Find the smallest element in $A[i..n]$. Swap it with $A[i]$.

These algorithms are easy to implement. But they run in $\Theta(n^2)$ time in the worst case.

4.2 Sorting in $O(n \log n)$ time

We have already seen that Mergesort sorts an array of numbers in $\Theta(n \log n)$ time. We will study two others: *Heapsort* and *Quicksort*.

4.2.1 Heaps

A *heap* is a left-complete binary tree that conforms to the *heap order*. The heap order property: in a (min) heap, for every node X , the key in the parent is smaller than or equal to the key in X . In other words, the parent node has key smaller than or equal to both of its children nodes. Similarly, in a max heap, the parent has a key larger than or equal both of its children. Thus the smallest key is in the root in a min heap; in the max heap, the largest is in the root.

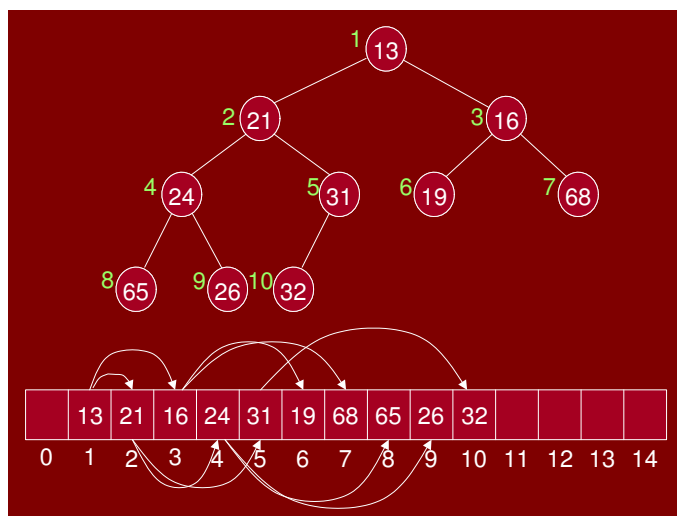


Figure 4.1: A min-heap

The number of nodes in a complete binary tree of height h is

$$n = 2^0 + 2^1 + 2^2 + \dots + 2^h = \sum_{i=0}^h 2^i = 2^{h+1} - 1$$

h in terms of n is

$$h = (\log(n + 1)) - 1 \approx \log n \in \Theta(\log n)$$

One of the clever aspects of heaps is that they can be stored in arrays without using any pointers. This is due to the left-complete nature of the binary tree. We store the tree nodes in level-order traversal. Access

to nodes involves simple arithmetic operations:

$\text{left}(i)$: returns $2i$, index of left child of node i .
 $\text{right}(i)$: returns $2i + 1$, the right child.
 $\text{parent}(i)$: returns $\lfloor i/2 \rfloor$, the parent of i .

The root is at position 1 of the array.

4.2.2 Heapsort Algorithm

We build a max heap out of the given array of numbers $A[1..n]$. We repeatedly extract the the maximum item from the heap. Once the max item is removed, we are left with a hole at the root. To fix this, we will replace it with the last leaf in tree. But now the heap order will very likely be destroyed. We will apply a heapify procedure to the root to restore the heap. Figure 4.2 shows an array being sorted.

```
HEAPSORT( array  $A$ , int  $n$ )
1  BUILD-HEAP( $A$ ,  $n$ )
2   $m \leftarrow n$ 
3  while ( $m \geq 2$ )
4  do SWAP( $A[1]$ ,  $A[m]$ )
5      $m \leftarrow m - 1$ 
6     HEAPIFY( $A$ , 1,  $m$ )
```

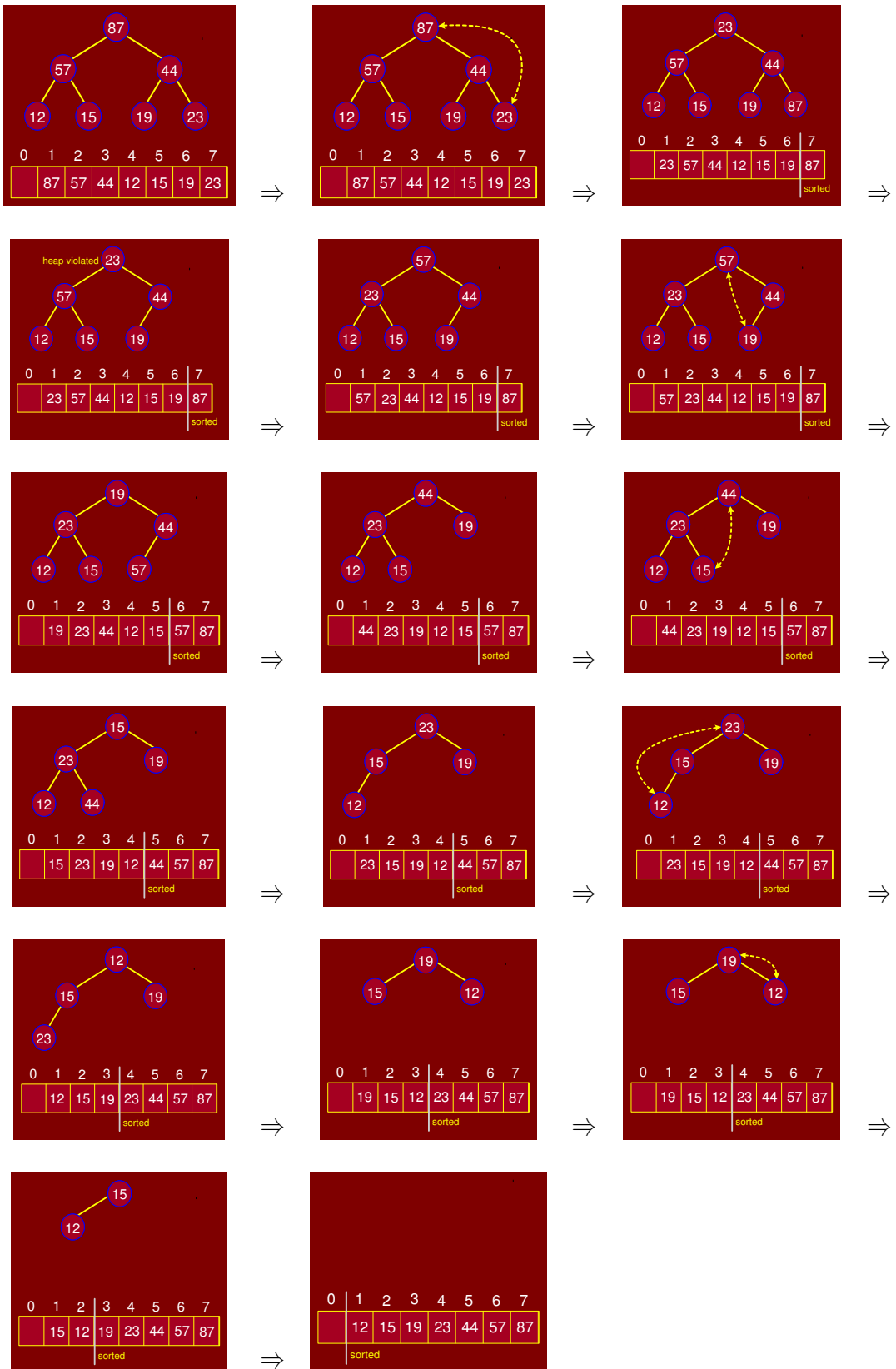


Figure 4.2: Example of heap sort

4.2.3 Heapify Procedure

There is one principal operation for maintaining the heap property. It is called Heapify. (In other books it is sometimes called sifting down.) The idea is that we are given an element of the heap which we suspect may not be in valid heap order, but we assume that all of other the elements in the subtree rooted at this element are in heap order. In particular this root element may be too small. To fix this we “sift” it down the tree by swapping it with one of its children. Which child? We should take the larger of the two children to satisfy the heap ordering property. This continues recursively until the element is either larger than both its children or until it falls all the way to the leaf level. Here is the algorithm. It is given the heap in the array A , and the index i of the suspected element, and m the current active size of the heap. The element $A[\max]$ is set to the maximum of $A[i]$ and its two children. If $\max \neq i$ then we swap $A[i]$ and $A[\max]$ and then recurse on $A[\max]$.

```

HEAPIFY( array A, int i, int m)
1  l ← LEFT(i)
2  r ← RIGHT(i)
3  max ← i
4  if (l ≤ m) and (A[l] > A[max])
5    then max ← l
6  if (r ≤ m) and (A[r] > A[max])
7    then max ← r
8  if (max ≠ i)
9    then SWAP(A[i], A[max])
10     HEAPIFY(A, max, m)

```

4.2.4 Analysis of Heapify

We call heapify on the root of the tree. The maximum levels an element could move up is $\Theta(\log n)$ levels. At each level, we do simple comparison which $O(1)$. The total time for heapify is thus $O(\log n)$. Notice that it is not $\Theta(\log n)$ since, for example, if we call heapify on a leaf, it will terminate in $\Theta(1)$ time.

4.2.5 BuildHeap

We can use Heapify to build a heap as follows. First we start with a heap in which the elements are not in heap order. They are just in the same order that they were given to us in the array A . We build the heap by starting at the leaf level and then invoke Heapify on each node. (Note: We cannot start at the top of the tree. Why not? Because the precondition which Heapify assumes is that the entire tree rooted at node i is already in heap order, except for i .) Actually, we can be a bit more efficient. Since we know that each leaf is already in heap order, we may as well skip the leaves and start with the first non-leaf node. This will be in position $n/2$. (Can you see why?)

Here is the code. Since we will work with the entire array, the parameter m for `Heapify`, which indicates the current heap size will be equal to n , the size of array A , in all the calls.

```
BUILDHEAP( array A, int n)
1  for i ← n/2 downto 1
2  do
3    HEAPIFY(A, i, n)
```

4.2.6 Analysis of BuildHeap

For convenience, we will assume $n = 2^{h+1} - 1$ where h is the height of tree. The heap is a left-complete binary tree. Thus at each level j , $j < h$, there are 2^j nodes in the tree. At level h , there will be 2^h or less nodes. How much work does `buildHeap` carry out? Consider the heap in Figure 4.3:

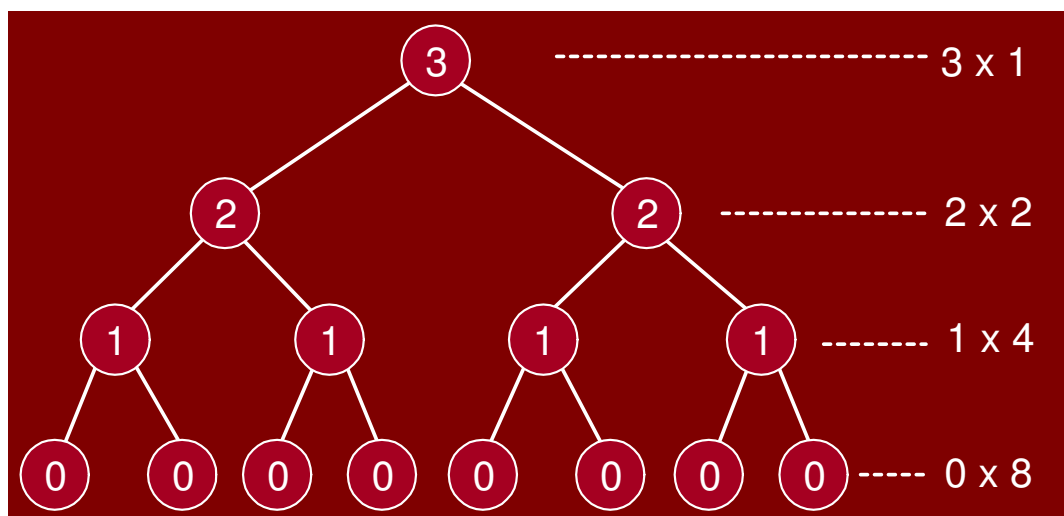


Figure 4.3: Total work performed in buildheap

At the bottom most level, there are 2^h nodes but we do not heapify these. At the next level up, there are 2^{h-1} nodes and each might shift down 1. In general, at level j , there are 2^{h-j} nodes and each may shift down j levels.

So, if count from bottom to top, level-by-level, the total time is

$$T(n) = \sum_{j=0}^h j2^{h-j} = \sum_{j=0}^h j \frac{2^h}{2^j}$$

We can factor out the 2^h term:

$$T(n) = 2^h \sum_{j=0}^h \frac{j}{2^j}$$

How do we solve this sum? Recall the geometric series, for any constant $x < 1$

$$\sum_{j=0}^{\infty} x^j = \frac{1}{1-x}$$

Take the derivative with respect to x and multiply by x

$$\sum_{j=0}^{\infty} jx^{j-1} = \frac{1}{(1-x)^2} \quad \sum_{j=0}^{\infty} jx^j = \frac{x}{(1-x)^2}$$

We plug $x = 1/2$ and we have the desired formula:

$$\sum_{j=0}^{\infty} \frac{j}{2^j} = \frac{1/2}{(1-(1/2))^2} = \frac{1/2}{1/4} = 2$$

In our case, we have a bounded sum, but since the infinite series is bounded, we can use it as an easy approximation:

$$\begin{aligned} T(n) &= 2^h \sum_{j=0}^h \frac{j}{2^j} \\ &\leq 2^h \sum_{j=0}^{\infty} \frac{j}{2^j} \\ &\leq 2^h \cdot 2 = 2^{h+1} \end{aligned}$$

Recall that $n = 2^{h+1} - 1$. Therefore

$$T(n) \leq n + 1 \in O(n)$$

The algorithm takes at least $\Omega(n)$ time since it must access every element at once. So the total time for BuildHeap is $\Theta(n)$.

BuildHeap is a relatively complex algorithm. Yet, the analysis yields that it takes $\Theta(n)$ time. An intuitive way to describe why it is so is to observe an important fact about binary trees. The fact is that the vast majority of the nodes are at the lowest level of the tree. For example, in a complete binary tree of height h , there is a total of $n \approx 2^{h+1}$ nodes.

The number of nodes at the bottom three levels alone is

$$2^h + 2^{h-1} + 2^{h-2} = \frac{n}{2} + \frac{n}{4} + \frac{n}{8} = \frac{7n}{8} = 0.875n$$

Almost 90% of the nodes of a complete binary tree reside in the 3 lowest levels. Thus, algorithms that operate on trees should be efficient (as BuildHeap is) on the bottom-most levels since that is where most of the weight of the tree resides.

4.2.7 Analysis of Heapsort

Heapsort calls BuildHeap once. This takes $\Theta(n)$. Heapsort then extracts roughly n maximum elements from the heap. Each extract requires a constant amount of work (swap) and $O(\log n)$ heapify. Heapsort is thus $O(n \log n)$.

Is HeapSort $\Theta(n \log n)$? The answer is yes. In fact, later we will show that comparison based sorting algorithms can not run faster than $\Omega(n \log n)$. Heapsort is such an algorithm and so is Mergesort that we saw earlier.

4.3 Quicksort

Our next sorting algorithm is Quicksort. It is one of the fastest sorting algorithms known and is the method of choice in most sorting libraries. Quicksort is based on the divide and conquer strategy. Here is the algorithm:

```

QUICKSORT( array A, int p, int r)
1  if (r > p)
2    then
3      i ← a random index from [p..r]
4      swap A[i] with A[p]
5      q ← PARTITION(A, p, r)
6      QUICKSORT(A, p, q - 1)
7      QUICKSORT(A, q + 1, r)

```

4.3.1 Partition Algorithm

Recall that the partition algorithm partitions the array $A[p..r]$ into three sub arrays about a pivot element x . $A[p..q - 1]$ whose elements are less than or equal to x , $A[q] = x$, $A[q + 1..r]$ whose elements are greater than x

We will choose the first element of the array as the pivot, i.e. $x = A[p]$. If a different rule is used for selecting the pivot, we can swap the chosen element with the first element. We will choose the pivot randomly.

The algorithm works by maintaining the following *invariant condition*. $A[p] = x$ is the pivot value. $A[p..q - 1]$ contains elements that are less than x , $A[q + 1..s - 1]$ contains elements that are greater than

or equal to x $A[s..r]$ contains elements whose values are currently unknown.

```

PARTITION( array A, int p, int r)
1   $x \leftarrow A[p]$ 
2   $q \leftarrow p$ 
3  for  $s \leftarrow p + 1$  to  $r$ 
4  do if ( $A[s] < x$ )
5      then  $q \leftarrow q + 1$ 
6          swap  $A[q]$  with  $A[s]$ 
7  swap  $A[p]$  with  $A[q]$ 
8  return  $q$ 

```

Figure 4.4 shows the execution trace partition algorithm.

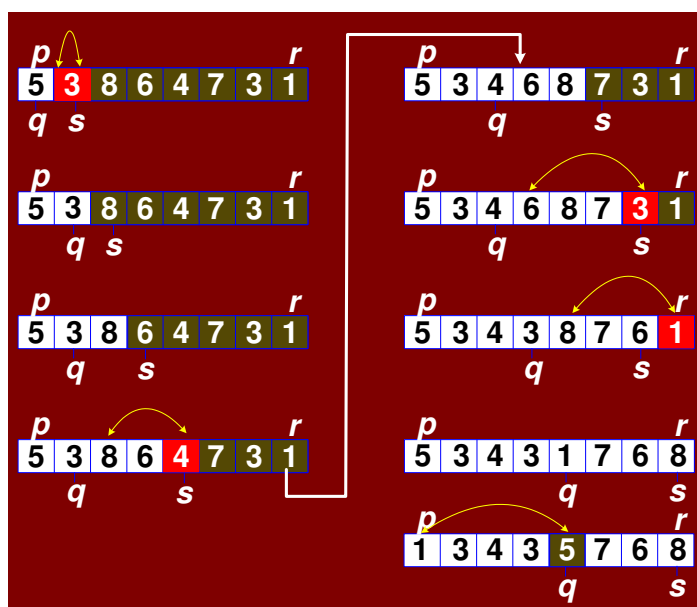


Figure 4.4: Trace of partitioning algorithm

4.3.2 Quick Sort Example

The Figure 4.5 trace out the quick sort algorithm. The first partition is done using the last element, 10, of the array. The left portion are then partitioned about 5 while the right portion is partitioned about 13. Notice that 10 is now at its final position in the eventual sorted order. The process repeats as the algorithm recursively partitions the array eventually sorting it.

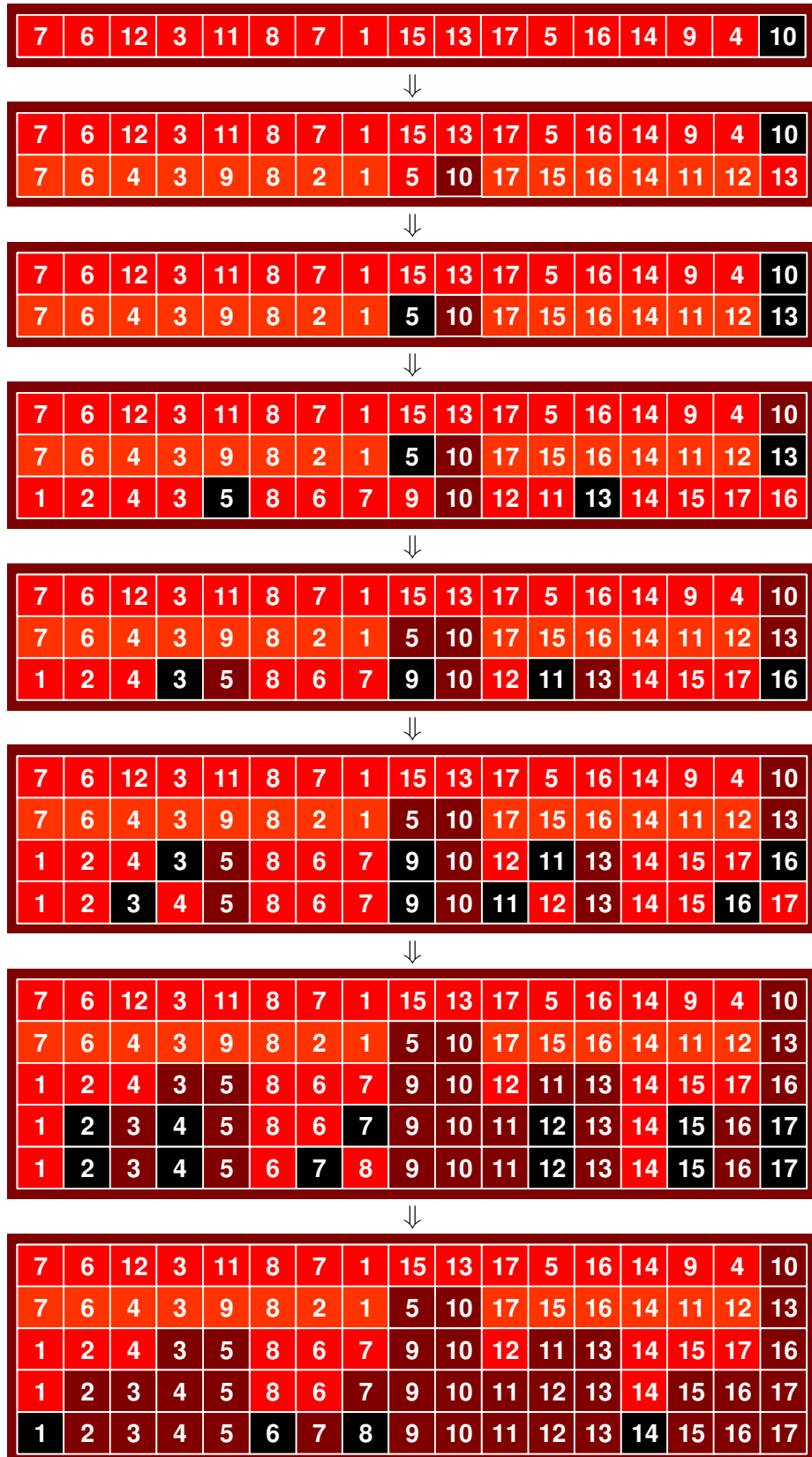


Figure 4.5: Example of quick sort

It is interesting to note (but not surprising) that the pivots form a binary search tree. This is illustrated in Figure 4.6.

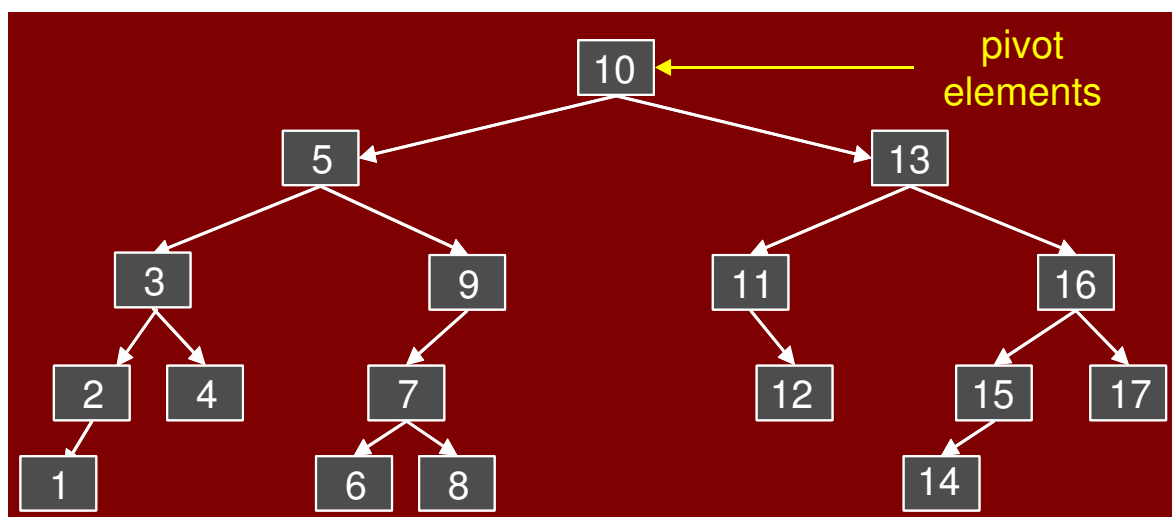


Figure 4.6: Quicksort BST

4.3.3 Analysis of Quicksort

The running time of quicksort depends heavily on the selection of the pivot. If the rank of the pivot is very large or very small then the partition (BST) will be unbalanced. Since the pivot is chosen randomly in our algorithm, the expected running time is $O(n \log n)$. The worst case time, however, is $O(n^2)$. Luckily, this happens rarely.

4.3.4 Worst Case Analysis of Quick Sort

Let's begin by considering the worst-case performance, because it is easier than the average case. Since this is a recursive program, it is natural to use a recurrence to describe its running time. But unlike MergeSort, where we had control over the sizes of the recursive calls, here we do not. It depends on how the pivot is chosen. Suppose that we are sorting an array of size n , $A[1 : n]$, and further suppose that the pivot that we select is of rank q , for some q in the range 1 to n . It takes $\Theta(n)$ time to do the partitioning and other overhead, and we make two recursive calls. The first is to the subarray $A[1 : q - 1]$ which has $q - 1$ elements, and the other is to the subarray $A[q + 1 : n]$ which has $n - q$ elements. So if we ignore the $\Theta(n)$ (as usual) we get the recurrence:

$$T(n) = T(q - 1) + T(n - q) + n$$

This depends on the value of q . To get the worst case, we maximize over all possible values of q . Putting is together, we get the recurrence

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ \max_{1 \leq q \leq n} (T(q-1) + T(n-q) + n) & \text{otherwise} \end{cases}$$

Recurrences that have max's and min's embedded in them are very messy to solve. The key is determining which value of q gives the maximum. (A rule of thumb of algorithm analysis is that the worst cases tends to happen either at the extremes or in the middle. So I would plug in the value $q = 1$, $q = n$, and $q = n/2$ and work each out.) In this case, the worst case happens at either of the extremes. If we expand the recurrence for $q = 1$, we get:

$$\begin{aligned} T(n) &\leq T(0) + T(n-1) + n \\ &= 1 + T(n-1) + n \\ &= T(n-1) + (n+1) \\ &= T(n-2) + n + (n+1) \\ &= T(n-3) + (n-1) + n + (n+1) \\ &= T(n-4) + (n-2) + (n-1) + n + (n+1) \\ &= T(n-k) + \sum_{i=1}^{k-2} (n-i) \end{aligned}$$

For the basis $T(1) = 1$ we set $k = n - 1$ and get

$$\begin{aligned} T(n) &\leq T(1) + \sum_{i=1}^{n-3} (n-i) \\ &= 1 + (3 + 4 + 5 + \dots + (n-1) + n + (n+1)) \\ &\leq \sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2} \in \Theta(n^2) \end{aligned}$$

4.3.5 Average-case Analysis of Quicksort

We will now show that in the average case, quicksort runs in $\Theta(n \log n)$ time. Recall that when we talked about average case at the beginning of the semester, we said that it depends on some assumption about the distribution of inputs. However, in the case of quicksort, the analysis does not depend on the distribution of input at all. It only depends upon the random choices of pivots that the algorithm makes. This is good, because it means that the analysis of the algorithm's performance is the same for all inputs. In this case the average is computed over all possible random choices that the algorithm might make for the choice of the pivot index in the second step of the QuickSort procedure above.

To analyze the average running time, we let $T(n)$ denote the average running time of QuickSort on a list of size n . It will simplify the analysis to assume that all of the elements are distinct. The algorithm has n

random choices for the pivot element, and each choice has an equal probability of $1/n$ of occurring. So we can modify the above recurrence to compute an average rather than a max, giving:

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ \frac{1}{n} \sum_{q=1}^n (T(q-1) + T(n-q) + n) & \text{otherwise} \end{cases}$$

The time $T(n)$ is the weighted sum of the times taken for various choices of q . I.e.,

$$\begin{aligned} T(n) = & \left[\frac{1}{n}(T(0) + T(n-1) + n) \right. \\ & + \frac{1}{n}(T(1) + T(n-2) + n) \\ & + \frac{1}{n}(T(2) + T(n-3) + n) \\ & \left. + \dots + \frac{1}{n}(T(n-1) + T(0) + n) \right] \end{aligned}$$

We have not seen such a recurrence before. To solve it, expansion is possible but it is rather tricky. We will attempt a constructive induction to solve it. We know that we want a $\Theta(n \log n)$. Let us assume that $T(n) \leq cn \log n$ for $n \geq 2$ where c is a constant.

For the base case $n = 2$ we have

$$\begin{aligned} T(2) &= \frac{1}{2} \sum_{q=1}^2 (T(q-1) + T(2-q) + 2) \\ &= \frac{1}{2} [(T(0) + T(1) + 2) + (T(1) + T(0) + 2)] \\ &= \frac{8}{2} = 4. \end{aligned}$$

We want this to be at most $c2 \log 2$, i.e.,

$$T(2) \leq c2 \log 2$$

or

$$4 \leq c2 \log 2$$

therefore

$$c \geq 4/(2 \log 2) \approx 2.88.$$

For the induction step, we assume that $n \geq 3$ and The induction hypothesis is that for any $n' < n$, we have $T(n') \leq cn' \log n'$. We want to prove that it is true for $T(n)$. By expanding $T(n)$ and moving the

factor of n outside the sum we have

$$\begin{aligned} T(n) &= \frac{1}{n} \sum_{q=1}^n (T(q-1) + T(n-q) + n) \\ &= \frac{1}{n} \sum_{q=1}^n (T(q-1) + T(n-q)) + n \\ &= \frac{1}{n} \sum_{q=1}^n T(q-1) + \frac{1}{n} \sum_{q=1}^n T(n-q) + n \end{aligned}$$

$$T(n) = \frac{1}{n} \sum_{q=1}^n T(q-1) + \frac{1}{n} \sum_{q=1}^n T(n-q) + n$$

Observe that the two sums add up the same values $T(0) + T(1) + \dots + T(n-1)$. One counts up and other counts down. Thus we can replace them with $2 \sum_{q=0}^{n-1} T(q)$. We will extract $T(0)$ and $T(1)$ and treat them specially. These two do not follow the formula.

$$\begin{aligned} T(n) &= \frac{2}{n} \left(\sum_{q=0}^{n-1} T(q) \right) + n \\ &= \frac{2}{n} \left(T(0) + T(1) + \sum_{q=2}^{n-1} T(q) \right) + n \end{aligned}$$

We will apply the induction hypothesis for $q < n$ we have

$$\begin{aligned} T(n) &= \frac{2}{n} \left(T(0) + T(1) + \sum_{q=2}^{n-1} T(q) \right) + n \\ &\leq \frac{2}{n} \left(1 + 1 + \sum_{q=2}^{n-1} (cq \log q) \right) + n \\ &= \frac{2c}{n} \left(\sum_{q=2}^{n-1} (cq \ln q) \right) + n + \frac{4}{n} \end{aligned}$$

We have never seen this sum before:

$$S(n) = \sum_{q=2}^{n-1} (cq \ln q)$$

Recall from calculus that for any monotonically increasing function $f(x)$

$$\sum_{i=a}^{b-1} f(i) \leq \int_a^b f(x) dx$$

The function $f(x) = x \ln x$ is monotonically increasing, and so

$$S(n) = \sum_{q=2}^{n-1} (cq \ln q) \leq \int_2^n x \ln x \, dx \quad (4.1)$$

We can integrate this by parts:

$$\int_2^n x \ln x \, dx = \frac{x^2}{2} \ln x - \frac{x^2}{4} \Big|_{x=2}^n$$

$$\begin{aligned} \int_2^n x \ln x \, dx &= \frac{x^2}{2} \ln x - \frac{x^2}{4} \Big|_{x=2}^n \\ &= \left(\frac{n^2}{2} \ln n - \frac{n^2}{4} \right) - (2 \ln 2 - 1) \\ &\leq \frac{n^2}{2} \ln n - \frac{n^2}{4} \end{aligned}$$

We thus have

$$S(n) = \sum_{q=2}^{n-1} (cq \ln q) \leq \frac{n^2}{2} \ln n - \frac{n^2}{4}$$

Plug this back into the expression for $T(n)$ to get

$$T(n) = \frac{2c}{n} \left(\frac{n^2}{2} \ln n - \frac{n^2}{4} \right) + n + \frac{4}{n}$$

$$\begin{aligned} T(n) &= \frac{2c}{n} \left(\frac{n^2}{2} \ln n - \frac{n^2}{4} \right) + n + \frac{4}{n} \\ &= cn \ln n - \frac{cn}{2} + n + \frac{4}{n} \\ &= cn \ln n + n \left(1 - \frac{c}{2} \right) + \frac{4}{n} \end{aligned}$$

$$T(n) = cn \ln n + n \left(1 - \frac{c}{2} \right) + \frac{4}{n}$$

To finish the proof, we want all of this to be at most $cn \ln n$. For this to happen, we will need to select c such that

$$n \left(1 - \frac{c}{2} \right) + \frac{4}{n} \leq 0$$

If we select $c = 3$, and use the fact that $n \geq 3$ we get

$$\begin{aligned} n \left(1 - \frac{c}{2} \right) + \frac{4}{n} &= \frac{3}{n} - \frac{n}{2} \\ &\leq 1 - \frac{3}{2} = -\frac{1}{2} \leq 0. \end{aligned}$$

From the basis case we had $c \geq 2.88$. Choosing $c = 3$ satisfies all the constraints. Thus $T(n) = 3n \ln n \in \Theta(n \log n)$.

4.4 In-place, Stable Sorting

An *in-place* sorting algorithm is one that uses no additional array for storage. A sorting algorithm is *stable* if duplicate elements remain in the same relative position after sorting.

9 3 3' 5 6 5' 2 1 3''	unsorted
1 2 3 3' 3'' 5 5' 6 9	stable sort
1 2 3' 3 3'' 5' 5 6 9	unstable

Bubble sort, insertion sort and selection sort are in-place sorting algorithms. Bubble sort and insertion sort can be implemented as stable algorithms but selection sort cannot (without significant modifications). Mergesort is a stable algorithm but not an in-place algorithm. It requires extra array storage. Quicksort is not stable but is an in-place algorithm. Heapsort is an in-place algorithm but is not stable.

4.5 Lower Bounds for Sorting

The best we have seen so far is $O(n \log n)$ algorithms for sorting. Is it possible to do better than $O(n \log n)$? If a sorting algorithm is solely based on comparison of keys in the array then it is *impossible* to sort more efficiently than $\Omega(n \log n)$ time. All algorithms we have seen so far are comparison-based sorting algorithms.

Consider sorting three numbers a_1, a_2, a_3 . There are $3! = 6$ possible combinations:

$$(a_1, a_2, a_3), (a_1, a_3, a_2), (a_3, a_2, a_1) \\ (a_3, a_1, a_2), (a_2, a_1, a_3), (a_2, a_3, a_1)$$

One of these permutations leads to the numbers in sorted order.

The comparison based algorithm defines a *decision tree*. Here is the tree for the three numbers.

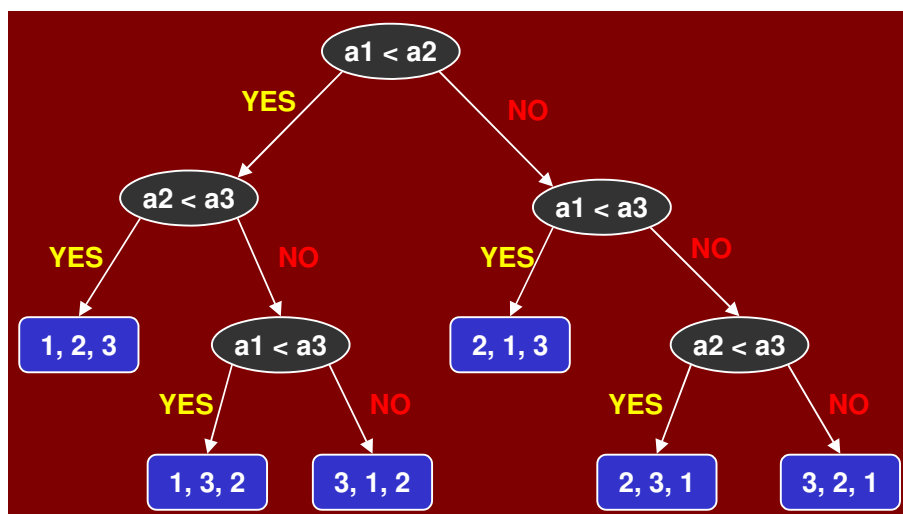


Figure 4.7: Decision Tree

For n elements, there will be $n!$ possible permutations. The height of the tree is exactly equal to $T(n)$, the running time of the algorithm. The height is $T(n)$ because any path from the root to a leaf corresponds to a sequence of comparisons made by the algorithm.

Any binary tree of height $T(n)$ has at most $2^{T(n)}$ leaves. Thus a comparison based sorting algorithm can distinguish between at most $2^{T(n)}$ different final outcomes. So we have

$$2^{T(n)} \geq n! \quad \text{and therefore} \\ T(n) \geq \log(n!)$$

We can use *Stirling's approximation* for $n!$:

$$n! \geq \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

Therefore

$$\begin{aligned} T(n) &\geq \log\left(\sqrt{2\pi n} \left(\frac{n}{e}\right)^n\right) \\ &= \log(\sqrt{2\pi n}) + n \log n - n \log e \\ &\in \Omega(n \log n) \end{aligned}$$

We thus have the following theorem.

Theorem 1

Any comparison-based sorting algorithm has worst-case running time $\Omega(n \log n)$.



CS502- Fundamentals of Algorithms
Solved MCQS
From Midterm Papers

May- 24 - 2013

MC100401285

Moaaz.pk@gmail.com

Mc100401285@vu.edu.pk

PSMD01

MIDTERM EXAMINATION
Fall 2011
CS502- Fundamentals of Algorithms

Question No: 1 (Marks: 1) - Please choose one

Due to left complete nature of binary tree, the heap can be stored in

- ▶ **Arrays (Page 40)**
- ▶ Structures
- ▶ Link Lis
- ▶ Stack

Question No: 1 (Marks: 1) - Please choose one

What type of instructions Random Access Machine (RAM) can execute?

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ **Arithmetic and logic (Page 10)**
- ▶ Parallel and recursive

Question No: 1 (Marks: 1) - Please choose one

For Chain Matrix Multiplication we can not use divide and conquer approach because,

- ▶ **We do not know the optimum k (Page 86)**
- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time
- ▶ Size of data is not given

Question No: 1 (Marks: 1) - Please choose one

What is the total time to heapify?

- ▶ **$O(\log n)$ (Page 43)**
- ▶ $O(n \log n)$
- ▶ $O(n^2 \log n)$
- ▶ $O(\log^2 n)$

Question No: 1 (Marks: 1) - Please choose one
word Algorithm comes from the name of the muslim author _____

▶ [Abu Ja'far Mohammad ibn Musa al-Khowarizmi.](#)

Question No: 1 (Marks: 1) - Please choose one
al-Khwarizmi's work was written in a book titled _____

▶ [al Kitab al-mukhtasar fi hisab al-jabr wa'l-muqabalah](#)

MIDTERM EXAMINATION
Spring 2010
CS502- Fundamentals of Algorithms

Question No: 1 (Marks: 1) - Please choose one
Random access machine or RAM is a/an

- ▶ Machine build by Al-Khwarizmi
- ▶ Mechanical machine
- ▶ Electronics machine
- ▶ [Mathematical model \(Page 10\)](#)

Question No: 2 (Marks: 1) - Please choose one
_____ is a graphical representation of an algorithm

- ▶ Σ notation
- ▶ Θ notation
- ▶ [Flowchart](#) [Click here for detail](#)
- ▶ Asymptotic notation

Question No: 3 (Marks: 1) - Please choose one
A RAM is an idealized machine with _____ random-access memory.

- ▶ 256MB
- ▶ 512MB
- ▶ [an infinitely large \(Page 10\)](#)
- ▶ 100GB

Question No: 4 (Marks: 1) - Please choose one

What type of instructions Random Access Machine (RAM) can execute? Choose best answer

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ **Arithmetic and logic (Rep)**
- ▶ Parallel and recursive

Question No: 5 (Marks: 1) - Please choose one

What will be the total number of max comparisons if we run brute-force maxima algorithm with n elements?

- ▶ n^2
- ▶ $\frac{n}{n^2}$
- ▶ **n (Page 14)**
- ▶ n^8

Question No: 6 (Marks: 1) - Please choose one

What is the solution to the recurrence $T(n) = T(n/2) + n$.

- ▶ $O(\log n)$
- ▶ **$O(n)$ (Page 37)**
- ▶ $O(n \log n)$
- ▶ $O(n^2)$

Question No: 7 (Marks: 1) - Please choose one

Consider the following code:

```
For(j=1; j<n; j++)
    For(k=1; k<15; k++)
        For(l=5; l<n; l++)
            {
                Do_something_constant();
            }
```

What is the order of execution for this code.

- ▶ **$O(n)$**
- ▶ $O(n^3)$
- ▶ $O(n^2 \log n)$
- ▶ $O(n^2)$

Question No: 8 (Marks: 1) - Please choose one

What is the total time to heapify?

- ▶ **$O(\log n)$ rep**
- ▶ $O(n \log n)$
- ▶ $O(n^2 \log n)$
- ▶ $O(\log^2 n)$

Question No: 9 (Marks: 1) - Please choose one

Consider the following Algorithm:

```
Factorial (n){  
  if (n=1)  
    return 1  
  else  
    return (n * Factorial(n-1))
```

Recurrence for the following algorithm is:

- ▶ $T(n) = T(n-1) + 1$
- ▶ $T(n) = nT(n-1) + 1$
- ▶ $T(n) = T(n-1) + n$
- ▶ $T(n) = T(n(n-1)) + 1$

Question No: 10 (Marks: 1) - Please choose one

When we call heapify then at each level the comparison performed takes time

- ▶ **It will take $\Theta(1)$ (Page 43)**
- ▶ Time will vary according to the nature of input data
- ▶ It can not be predicted
- ▶ It will take $\Theta(\log n)$

Question No: 11 (Marks: 1) - Please choose one

In Quick sort, we don't have the control over the sizes of recursive calls

- ▶ **True (Page 40)**
- ▶ False
- ▶ Less information to decide
- ▶ Either true or false

Question No: 12 (Marks: 1) - Please choose one

Is it possible to sort without making comparisons?

- ▶ **Yes (Page 57)**
- ▶ No

Question No: 13 (Marks: 1) - Please choose one

If there are $\Theta(n^2)$ entries in edit distance matrix then the total running time is

- ▶ $\Theta(1)$
- ▶ **$\Theta(n^2)$ [Click here for detail](#)**
- ▶ $\Theta(n)$
- ▶ $\Theta(n \log n)$

Question No: 14 (Marks: 1) - Please choose one

For Chain Matrix Multiplication we can not use divide and conquer approach because,

- ▶ **We do not know the optimum k (Page 86)**
- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time
- ▶ Size of data is not given

Question No: 15 (Marks: 1) - Please choose one

The Knapsack problem belongs to the domain of _____ problems.

- ▶ **Optimization (Page 91)**
- ▶ NP Complete
- ▶ Linear Solution
- ▶ Sorting

Question No: 16 (Marks: 1) - Please choose one

Suppose we have three items as shown in the following table, and suppose the capacity of the knapsack is 50 i.e. $W = 50$.

Item	Value	Weight
1	60	10
2	100	20
3	120	30

The optimal solution is to pick

- ▶ Items 1 and 2
- ▶ Items 1 and 3
- ▶ **Items 2 and 3 (correct)**
- ▶ None of these

MIDTERM EXAMINATION
Spring 2010
CS502- Fundamentals of Algorithms

Question No: 1 (Marks: 1) - Please choose one

For the Sieve Technique we take time

- ▶ **T(nk) (Page 34)**
- ▶ T(n / 3)
- ▶ n²
- ▶ n/3

Question No: 1 (Marks: 1) - Please choose one

Sieve Technique applies to problems where we are interested in finding a single item from a larger set of _____

Select correct option:

- ▶ **n items (Page 34)**
- ▶ phases
- ▶ pointers
- ▶ constant

Question No: 1 (Marks: 1) - Please choose one

_____ graphical representation of algorithm.

- ▶ asymptotic
- ▶ **Flowchart (rep)**

Question No: 1 (Marks: 1) - Please choose one

who invented the quick sort

- ▶ **C.A.R. Hoare** [Click here for detail](#)

Question No: 1 (Marks: 1) - Please choose one

main elements to a divide-and-conquer

- ▶ **Divide, conquer, combine (Page 27)**

Question No: 1 (Marks: 1) - Please choose one

Mergesort is a stable algorithm but not an in-place algorithm.

- ▶ **True (Page 54)**
- ▶ false

Question No: 1 (Marks: 1) - Please choose one

Counting sort the numbers to be sorted are in the range 1 to k where k is small.

- ▶ **True (Page 57)**
- ▶ False

MIDTERM EXAMINATION
Spring 2007
CS502- Fundamentals of Algorithms

Question No: 1 (Marks: 1) - Please choose one

Total time for heapify is:

- ▶ $O(\log_2 n)$
- ▶ $O(n \log_2 n)$
- ▶ $O(n^2 \log_2 n)$
- ▶ **$O(\log_2 n)$ Rep**

Question No: 1 (Marks: 1) - Please choose one

If an algorithm has a complexity of $\log_2 n + n \log_2 n + n$. we could say that it has complexity

- ▶ $O(n)$
- ▶ $O(n \log_2 n)$
- ▶ $O(3)$
- ▶ $O(\log_2 (\log_2 n))$
- ▶ $O(\log_2 n)$

Question No: 1 (Marks: 1) - Please choose one

In RAM model instructions are executed

- ▶ **One after another (Page 10)**
- ▶ Parallel
- ▶ Concurrent
- ▶ Random

Question No: 1 (Marks: 1) - Please choose one

In selection algorithm, because we eliminate a constant fraction of the array with each phase, we get the

- ▶ **Convergent geometric series (Page 37)**
- ▶ Divergent geometric series
- ▶ None of these

Question No: 1 (Marks: 1) - Please choose one

Due to left-complete nature of binary tree, heaps can be stored in

- ▶ Link list
- ▶ Structure
- ▶ **Array (Page 40)**
- ▶ None of above

CS609- System Programming Midterm Quizzes (Quiz No.1 & 2)

Quiz No.1 (04 – MAY - 2013)

Question No: 1 (Marks: 1) - Please choose one

The time assumed for each basic operation to execute on RAM model of computation is-----

- Infinite
- Continuous
- Constant (Page 10)**
- Variable

Question No: 1 (Marks: 1) - Please choose one

If the indices passed to merge sort algorithm are not equal, the algorithm may return immediately.

- True
- False (Page 28)**

Question No: 1 (Marks: 1) - Please choose one

Brute-force algorithm uses no intelligence in pruning out decisions.

- True (Page 18)**
- False

Question No: 1 (Marks: 1) - Please choose one

In analysis, the Upper Bound means the function grows asymptotically no faster than its largest term.

True (Page 24)

False

Question No: 1 (Marks: 1) - Please choose one

For small values of n, any algorithm is fast enough. Running time does become an issue when n gets large.

True (Page 14)

Fast

Question No: 1 (Marks: 1) - Please choose one

The array to be sorted is not passed as argument to the merge sort algorithm.

True

False

Question No: 1 (Marks: 1) - Please choose one

In simple brute-force algorithm, we give no thought to efficiency.

True (Page 11)

False

Question No: 1 (Marks: 1) - Please choose one

The ancient Roman politicians understood an important principle of good algorithm design that is plan-sweep algorithm.

True

False (Page 27) [Divide and Conquer]

Question No: 1 (Marks: 1) - Please choose one

In 2d-space a point is said to be _____ if it is not dominated by any other point in that space.

Member

Minimal

Maximal (Page 11)

Joint

Question No: 1 (Marks: 1) - Please choose one

An algorithm is a mathematical entity that is dependent on a specific programming language.

True

False (Page 7)

Question No: 1 (Marks: 1) - Please choose one

The running time of an algorithm would not depend upon the optimization by the compiler but that of an implementation of the algorithm would depend on it.

True (Page 13)

False

Question No: 1 (Marks: 1) - Please choose one

$F(n)$ and $g(n)$ are asymptotically equivalent. This means that they have essentially the same _____ for large n .

Results

Variables

Size

Growth rates (Page 23)

Question No: 1 (Marks: 1) - Please choose one

$8n^2 + 2n - 3$ will eventually exceed $c2^*(n)$ no matter how large we make $c2$.

True (Page 25)

False

Question No: 1 (Marks: 1) - Please choose one

If we associate (x, y) integers pair to cars where x is the speed of the car and y is the negation of the price. High y value for a car means a _____ car.

Fast

Slow

Expensive

Cheap (Page 11)

Question No: 1 (Marks: 1) - Please choose one

The function $f(n) = n(\log n + 1)/2$ is asymptotically equivalent to $n \log n$. Here Upper Bound means the function $f(n)$ grows asymptotically _____ faster than $n \log n$.

More

Quiet

Not (Page 24)

At least

Question No: 1 (Marks: 1) - Please choose one

After sorting in merge sort algorithm, merging process is invoked.

Select correct option:

True (Page 28)

False

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 (Marks: 1) - Please choose one

Asymptotic growth rate of the function is taken over _____ case running time.

Select correct option:

Best

Average

Worst (Page 14)

Normal

Question No: 1 (Marks: 1) - Please choose one

In analysis of $f(n) = n(n/5) + n - 10 \log n$, $f(n)$ is asymptotically equivalent to _____.

n

$2n$

$n+1$

n^2 (Page 23)

Question No: 1 (Marks: 1) - Please choose one

Algorithm is concerned with.....issues.

Macro

Micro

Both Macro & Micro (Page 8)

Normal

Question No: 1 (Marks: 1) - Please choose one

We cannot make any significant improvement in the running time which is better than that of brute-force algorithm.

True

False (Page 18)

Question No: 1 (Marks: 1) - Please choose one

In addition to passing in the array itself to Merge Sort algorithm, we will pass in _____ other arguments which are indices.

Two (Page 28)

Three

Four

Five

Question No: 1 (Marks: 1) - Please choose one

Consider the following Algorithm: $\text{Fun}(n) \{ \text{if } (n=1) \text{ return } 1 \text{ else return } (n * \text{Fun}(n-1)) \}$ Recurrence for the above algorithm is:

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

$nT(n-1)+1$
 $2T(n-1)+1$
 $T(n-1)+cn$
 $T(n-1)+1$

Question No: 1 (Marks: 1) - Please choose one

In analysis, the Lower Bound means the function grows asymptotically at least as fast as its largest term.

True (Page 24)

False

Question No: 1 (Marks: 1) - Please choose one

Efficient algorithm requires less computational.....

Memory

Running Time

Memory and Running Time (Page 9)

Energy

Question No: 1 (Marks: 1) - Please choose one

The O-notation is used to state only the asymptotic _____ bounds.

Two

Lower

Upper (Page 25)

Both lower & upper

Question No: 1 (Marks: 1) - Please choose one

For the worst-case running time analysis, the nested loop structure containing one “for” and one “while” loop, might be expressed as a pair of _____ nested summations.

1

2 (Page 16)

3

4

Question No: 1 (Marks: 1) - Please choose one

Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their _____ coordinates.

X (Page 18)

Y

Z

X & Y

Question No: 1 (Marks: 1) - Please choose one

Brute-force algorithm for 2D-Maxima is operated by comparing _____ pairs of points.

- Two
- Some
- Most

All (Page 18)

Question No: 1 (Marks: 1) - Please choose one

The function $f(n)=n(\log n+1)/2$ is asymptotically equivalent to $n \log n$. Here Lower Bound means function $f(n)$ grows asymptotically at _____ as fast as $n \log n$.

- Normal
- Least (Page 23)
- Most
- All

Question No: 1 (Marks: 1) - Please choose one

The definition of Theta-notation relies on proving _____ asymptotic bound.

- One
- Lower
- Upper

Both lower & upper (Page 25) rep

Question No: 1 (Marks: 1) - Please choose one

In plane sweep approach, a vertical line is swept across the 2d-plane and _____ structure is used for holding the maximal points lying to the left of the sweep line.

- Array
- Queue
- Stack (Page 18)
- Tree

Question No: 1 (Marks: 1) - Please choose one

Algorithm analysts know for sure about efficient solutions for NP-complete problems.

Select correct option:

- True

False (Page 9)

Quiz No.1 (2012)

Question No: 1 of 10 (Marks: 1) - Please choose one

The number of nodes in a complete binary tree of height h is

$2^{(h+1)} - 1$ (Page 40)

$2 * (h+1) - 1$

$2 * (h+1)$

$((h+1) ^ 2) - 1$

Question No: 1 of 10 (Marks: 1) - Please choose one

The analysis of Selection algorithm shows the total running time is indeed _____ in n,

arithmetic

geometric

linear (Page 37)

orthogonal

Question No: 1 of 10 (Marks: 1) - Please choose one

A (an) _____ is a left-complete binary tree that conforms to the heap order

heap (Page 40)

binary tree

binary search tree

array

Question No: 1 of 10 (Marks: 1) - Please choose one

Analysis of Selection algorithm ends up with,

$T(n)$ (Page 37)

$T(1 / 1 + n)$

$T(n / 2)$

$T((n / 2) + n)$

Question No: 1 of 10 (Marks: 1) - Please choose one

For the sieve technique we solve the problem,

recursively (Page 34)

mathematically

precisely

accurately

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

A heap is a left-complete binary tree that conforms to the _____

increasing order only

decreasing order only

heap order (Page 40)

(log n) order

Question No: 1 of 10 (Marks: 1) - Please choose one

In which order we can sort?

increasing order only

decreasing order only

increasing order or decreasing order (Page 39)

both at the same time

Question No: 1 of 10 (Marks: 1) - Please choose one

Divide-and-conquer as breaking the problem into a small number of

pivot

Sieve

smaller sub problems (Page 34)

Selection

Question No: 1 of 10 (Marks: 1) - Please choose one

For the heap sort we store the tree nodes in

level-order traversal (Page 40)

in-order traversal

pre-order traversal

post-order traversal

Question No: 1 of 10 (Marks: 1) - Please choose one

The sieve technique works in _____ as follows

Phases (Page 34)

numbers

integers

routines

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

CS502 - Fundamentals of Algorithms

Quiz No.1 12-11-2012

Question No: 1 of 10 (Marks: 1) - Please choose one

We do sorting to,

keep elements in random positions

keep the algorithm run in linear order

keep the algorithm run in $(\log n)$ order

keep elements in increasing or decreasing order

Question No: 1 of 10 (Marks: 1) - Please choose one

Heaps can be stored in arrays without using any pointers; this is due to the _____ nature of the binary tree,

left-complete (Page 40)

right-complete

tree nodes

tree leaves

Question No: 1 of 10 (Marks: 1) - Please choose one

Sieve Technique can be applied to selection problem?

True (Page 35)

False

Question No: 1 of 10 (Marks: 1) - Please choose one

In Sieve Technique we do not know which item is of interest

True (Page 34)

False

Question No: 1 of 10 (Marks: 1) - Please choose one

In the analysis of Selection algorithm, we eliminate a constant fraction of the array with each phase; we get the convergent _____ series in the analysis,

linear

arithmetic

geometric (Page 37)

exponent

Question No: 1 of 10 (Marks: 1) - Please choose one

For the heap sort, access to nodes involves simple _____ operations.

arithmetic (Page 41)

binary
algebraic
logarithmic

Question No: 1 of 10 (Marks: 1) - Please choose one

Slow sorting algorithms run in,

$T(n^2)$ (Page 39)

$T(n)$
 $T(\log n)$

Question No: 1 of 10 (Marks: 1) - Please choose one

In the analysis of Selection algorithm, we make a number of passes, in fact it could be as many as,

$T(n)$
 $T(n/2)$

$\log n$ (Page 37)

$n/2 + n/4$

Question No: 1 of 10 (Marks: 1) - Please choose one

The sieve technique is a special case, where the number of sub problems is just

5
many

1 (Page 34)

few

Question No: 1 of 10 (Marks: 1) - Please choose one

How many elements do we eliminate in each time for the Analysis of Selection algorithm?

$(n/2)+n$ elements

$(n/2)$ elements (Page 37)

$n/4$ elements

$2n$ elements

Question No: 1 of 10 (Marks: 1) - Please choose one

One of the clever aspects of heaps is that they can be stored in arrays without using any _____.

pointers (Page 40)

constants
variables
functions

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

How much time merge sort takes for an array of numbers?

$T(n^2)$

$T(n)$

$T(\log n)$

$T(n \log n)$ (Page 40)

Question No: 1 of 10 (Marks: 1) - Please choose one

The reason for introducing Sieve Technique algorithm is that it illustrates a very important special case of, **divide-and-conquer (Page 34)**

decrease and conquer

greedy nature

2-dimension Maxima

Question No: 1 of 10 (Marks: 1) - Please choose one

In Sieve Technique we do not know which item is of interest

True (Page 34) rep

False

Question No: 1 of 10 (Marks: 1) - Please choose one

Theta asymptotic notation for $T(n)$:

Set of functions described by: $c_1g(n)$ Set of functions described by $c_1g(n) \geq f(n)$ for c_1 s

Theta for $T(n)$ is actually upper and worst case comp (Not sure)

Set of functions described by:

$c_1g(n)$

Question No: 1 of 10 (Marks: 1) - Please choose one

Memoization is?

To store previous results for future use

To avoid this unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later (page 74)

To make the process accurate

None of the above

Question No: 1 of 10 (Marks: 1) - Please choose one

Which sorting algorithm is faster

$O(n \log n)$ Page 26

$O(n^2)$

$O(n+k)$

$O(n^3)$

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

Quick sort is

Stable & in place

Not stable but in place (Page 54)

Stable but not in place

Some time stable & some times in place

Question No: 1 of 10 (Marks: 1) - Please choose one

One example of in place but not stable algorithm is

Merger Sort

Quick Sort (Page 54)

Continuation Sort

Bubble Sort

Question No: 1 of 10 (Marks: 1) - Please choose one

Cont sort is suitable to sort the elements in range 1 to k

K is Large

K is not known

K may be small or large

K is small (Page 57)

Question No: 1 of 10 (Marks: 1) - Please choose one

In place stable sorting algorithm.

If duplicate elements remain in the same relative position after sorting (Page 54)

One array is used

More than one arrays are required

Duplicating elements not handled

Question No: 1 of 10 (Marks: 1) - Please choose one

Which may be a stable sort?

Merger

Insertion (Page 54)

Both above

None of the above

Question No: 1 of 10 (Marks: 1) - Please choose one

An in place sorting algorithm is one that uses ___ arrays for storage

Two dimensional arrays

More than one array

No Additional Array (Page 54)

None of the above

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

Sieve Technique applies to problems where we are interested in finding a single item from a larger set of

n items (Page 34)

phases
pointers
constant

Question No: 1 of 10 (Marks: 1) - Please choose one

Sorting is one of the few problems where provable _____ bounds exists on how fast we can sort,

upper

lower (Page 39)

average
log n

Question No: 1 of 10 (Marks: 1) - Please choose one

Counting sort has time complexity:

O(n) (Page 58)

O(n+k)
O(k)
O(nlogn)

Question No: 1 of 10 (Marks: 1) - Please choose one

The running time of quick sort depends heavily on the selection of

No of inputs

Arrangement of elements in array

Size o elements

Pivot elements (Page 49)

Question No: 1 of 10 (Marks: 1) - Please choose one

Which may be stable sort:

Bubble sort

Insertion sort

Both of above (Page 54)

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Moderen Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

One Example of in place but not stable sort is

Quick (Page 54)

Heap

Merge

Bubble

Question No: 1 of 10 (Marks: 1) - Please choose one

In Quick Sort Constants hidden in $T(n \log n)$ are

Large

Medium

Small [Click here for detail](#)

Not Known

Question No: 1 of 10 (Marks: 1) - Please choose one

Quick sort is based on divide and conquer paradigm; we divide the problem on base of pivot element and:

There is explicit combine process as well to conquer the solution.

No work is needed to combine the sub-arrays, the array is already sorted

Merging the sub arrays

None of above. (Page 51)

Ref: - random choices for the pivot element and each choice have an equal probability of $1/n$ of occurring. So we can modify the above recurrence to compute an average rather than a max

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

CS501 - Quiz No.2 (Spring 2013)

Question No: 1 of 10 (Marks: 1) - Please choose one

A point p in 2-dimensional space is usually given by its integer coordinate(s)_____

p.x only

p.y only

p.x & p.z

p.x & p.y (Page 10)

Question No: 1 of 10 (Marks: 1) - Please choose one

In _____ we have to find rank of an element from given input.

Merge sort algorithm

Selection problem (Page 34)

Brute force technique

Plane Sweep algorithm

Question No: 1 of 10 (Marks: 1) - Please choose one

In Heap Sort algorithm, if heap property is violated _____

We call Build heap procedure

We call Heapify procedure

We ignore

Heap property can never be violated

Question No: 1 of 10 (Marks: 1) - Please choose one

Upper bound requires that there exist positive constants c_2 and n_0 such that $f(n) \leq c_2n$ for all $n \leq n_0$ (ye question ghalat lag raha hai mujhae

Less than

Equal to or Less than (Page 25)

Equal or Greater than

Greater than

Question No: 1 of 10 (Marks: 1) - Please choose one

A RAM is an idealized algorithm with takes an infinitely large random-access memory.

True

False (Page 10)

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

_____ is one of the few problems, where provable lower bounds exist on how fast we can sort.

Searching

Sorting (Page)

Both Searching & Sorting

Graphing

Question No: 1 of 10 (Marks: 1) - Please choose one

Floor and ceiling are _____ to calculate while analyzing algorithms.

Very easy

Usually considered difficult (Page 31)

Question No: 1 of 10 (Marks: 1) - Please choose one

In Heap Sort algorithm, the maximum levels an element can move upward is _____

Theta (log n) (Page 43)

Order (log n)

Omega (log n)

O (1) i.e. Constant time

Question No: 1 of 10 (Marks: 1) - Please choose one

A point p in 2-dimensional space is usually given by its integer coordinate(s)_____

p.x only p.y

only p.x & p.z

p.x & p.y (Page 17)

Question No: 1 of 10 (Marks: 1) - Please choose one

In Heap Sort algorithm, the total running time for Heapify procedure is _____

Theta (log n) (Page 43)

Order (log n)

Omega (log n)

O (1) i.e. Constant time

Question No: 1 of 10 (Marks: 1) - Please choose one

Algorithm is a mathematical entity, which is independent of a specific machine and operating system.

True

False (Page 7)

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

While Sorting, the ordered domain means for any two input elements x and y _____ satisfies only.

x < y

x > y

x = y

All of the above (Page 39)

Question No: 1 of 10 (Marks: 1) - Please choose one

Quick sort is best from the perspective of Locality of reference.

True (Page 9)

False

Question No: 1 of 10 (Marks: 1) - Please choose one

Sorting can be in _____

Increasing order only

Decreasing order only

Both Increasing and Decreasing order (Page 39)

Random order

Question No: 1 of 10 (Marks: 1) - Please choose one

In Heap Sort algorithm, we build _____ for ascending sort.

Max heap (Page 41)

Min heap

Question No: 1 of 10 (Marks: 1) - Please choose one

In Sieve Technique, we know the item of interest.

True

False (Page 34)

Question No: 1 of 10 (Marks: 1) - Please choose one

While solving Selection problem, in Sieve technique we partition input data _____

In increasing order

In decreasing order

According to Pivot (Page 35)

Randomly

Muhammad Moaaz Siddiq – MCS(4th)

Moaaz.pk@gmail.com

**Campus:- Institute of E-Learning & Modern Studies
(IEMS) Samundari**

Question No: 1 of 10 (Marks: 1) - Please choose one

In pseudo code, the level of details depends on intended audience of the algorithm.

True (Page 12)

False

Question No: 1 of 10 (Marks: 1) - Please choose one

The sieve technique works where we have to find _____ item(s) from a large input.

Single (Page 34)

Two

Three

Similar

Question No: 1 of 10 (Marks: 1) - Please choose one

If the indices passed to merge sort algorithm are _____, then this means that there is only one element to sort.

Small

Large

Equal (Page 28)

Not Equal



CS-502 Fundamentals Of Algorithms
Update MCQS For Mid Term
Solve By Vu Topper RM



80 To 100% Marks



وَتَعَزُّ مِنْ تَشَاءٍ وَتَذَلُّ مِنْ تَشَاءٍ



PROFESSIONAL ONLINE ACADEMY

We Offers

LMS Handling

Important Notes

Online Classes

Assignments

Quiz & GDB's

Projects

**NOTHING
IS
IMPOSSIBLE**

**Join Us
Now**

For More Info

Contact us at:

Rizwan Manzoor

0322-4021365



بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No: 01

(Marks:1)

Vu-Topper RM

There are _____ entries in the Edit Distance Matrix.

A. $\Theta(n)$

B. $\Theta(n^2)$

Page 84

C. $\Theta(n+2)$

D. $\Theta(n + 100)$

Question No: 02

(Marks:1)

Vu-Topper RM

For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from _____

A. All possible random values

Page 50

B. Pivot is input separately

C. Values greater than 5

Question No: 03

(Marks:1)

Vu-Topper RM

As per algorithm of Dynamic Programming, we need to store

A. First sub-problem only

B. Best solution only

C. Intermediate sub-problems

Page 75

D. Final solution only

Question No: 04

(Marks:1)

Vu-Topper RM

In chain matrix multiplication, table is filled _____ to find the multiplication of matrix.

A. row wise

B. column wise

C. diagonally

D. bottom-to-up

Page 86

Question No: 05

(Marks:1)

Vu-Topper RM

The only way to convert a string of i characters into the empty string is with i deletions, represented as

A. $E(0,j) = j$

B. $E(i,j) = 1$

C. $E(0,i) = j$

D. $E(i,0) = i$

Page 78

Question No: 06

(Marks:1)

Vu-Topper RM

If there are $\theta(n^2)$ entries in edit distance matrix then the total running time is:

A. $\theta(n)$

B. $\theta(1)$

C. $\theta(n^2)$

Page 84

D. $\theta(n \log n)$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No: 07

(Marks:1)

Vu-Topper RM

In average –case time analysis of quick sort algorithm , the most balanced case for partition is when we divide the list of elements into _.

A. Equal no. of pieces as of input elements

- B. Single piece exactly
- C. Two nearly equal pieces
- D. Three nearly equal pieces

Question No: 08

(Marks:1)

Vu-Topper RM

If matrix A of dimension $p \times q$ is multiply with matrix B of dimension $q \times r$, then each entry in resultant matrix takes _____ time.

A. O (q)

Page 84

- B. (1)
- C. (p x q)
- D. (q x r)

Question No: 09

(Marks:1)

Vu-Topper RM

Fibonacci Sequence was named on _____, a famous mathematician in 12th Century.

- A. Fred Brooks
- B. Grady Booch

C. Leonardo Pisano

Page 73

- D. Edgar F. Codd

Question No: 10

(Marks:1)

Vu-Topper RM

In quick sort algorithm, we choose pivot_____.

- A. Always the smallest element
- B. Greater than 5

C. Randomly

Page 35

- D. Less than 5

Question No: 11

(Marks:1)

Vu-Topper RM

For comparison-based sorting algorithms, it is possible to sort more efficiently than $\Omega(n \log(n))$ time.

- A. Always
- B. Sometimes not

C. NOT

Page 54

- D. Sometimes

Question No:12

(Marks:1)

Vu-Topper RM

The sequence of merge sort algorithm is:

- A. Divide Combine-Conquer
- B. Conquer-Divide-Combine

C. Divide-Conquer-Combine

Page 27

- D. Combine-Divide-Conquer

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No: 13

(Marks:1)

Vu-Topper RM

In _____ Knapsack Problem, limitation is that an item can either be put in the bag or not. Fractional items are not allowed.

A. 0

B. 1

C. 0/1

D. Fractional

Page 91

Question No: 14

(Marks:1)

Vu-Topper RM

In Selection algorithm, we assume pivot selection takes theta _____ running time.

A. n

B. n^2

C. n^3

D. $\log(n)$

Page 36

Question No: 15

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm (using max heap), when every time maximum elements removed from top _____.

A. We call merge Sort Algorithm

B. it becomes Order n^2 Algorithm

C. Divide and Conquer strategy helps us

D. We are left with a hole

Page 41

Question No: 16

(Marks:1)

Vu-Topper RM

_____ is a method of solving a problem in which we check all possible solutions to the problem to find the solution we need.

A. Plane-Sweep Algorithm

B. Sorting Algorithm

C. Brute-Force Algorithm

D. Greedy approach

Google

Question No: 17

(Marks:1)

Vu-Topper RM

The worst case running time of Quick sort algorithm _____.

A. Is quadratic

B. Is linear

C. Cannot be quadratic

D. Is always Exponential

Question No: 18

(Marks:1)

Vu-Topper RM

In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with _____ leaf in the tree.

A. Last

B. First

C. Any

Page 41

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. Second last

Question No: 19

(Marks:1)

Vu-Topper RM

Quick sort algorithm was developed by –

- A. AlferdAho
- B. Sedgewick
- C. John Vincent Atanasoff

D. Tony Hoare

Google

Question No: 20

(Marks:1)

Vu-Topper RM

If Matrix-A has dimensions “ 3×2 ” and Matrix-B has dimensions “ 2×3 ”, then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.

- A. 3×2
- B. 2×3
- C. 2×2

D. 3×3

Question No: 21

(Marks:1)

Vu-Topper RM

In Sorting the key value or attribute _____ from an ordered domain.

A. Must be

Page 39

- B. Not always
- C. May be
- D. Occasionally

Question No: 22

(Marks:1)

Vu-Topper RM

Result of asymptotical analysis of $n(n-3)$ and $4n*n$ is that _____

- A. $n(n-1)$ is asymptotically Less
- B. $n(n-1)$ is asymptotically Greater
- C. Both are asymptotically Not equivalent

D. Both are asymptotically Equivalent

Page 23

Question No: 23

(Marks:1)

Vu-Topper RM

Floor and ceiling are _____ to calculate while analyzing algorithms

- A. Very easy
- B. 3rd Option is missing
- C. Usually considered difficult**
- D. 4th Option is missing

Question No: 24

(Marks:1)

Vu-Topper RM

_____ of reference is an important fact of current processor technology.

- A. Defining
- B. Assigning
- C. Locality**
- D. Formality

Page 8

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No: 25

(Marks:1)

Vu-Topper RM

In max-heap, largest element is stored at root node. Where is the smallest element stored?

A. Right Node

B. Leaf Node

C. Middle Node

D. Left Node

Question No: 26

(Marks:1)

Vu-Topper RM

Which of the following is calculated with Big Omega notation?

A. Medium bounds

B. Upper bounds

C. Lower bounds **Page 25**

D. Both upper and lower bounds

Question No: 27

(Marks:1)

Vu-Topper RM

Edit distance algorithm based on _____ strategy

A. Greedy

B. Dynamic Programming **Page 81**

C. Divide and Conquer

D. Searching

Question No: 28

(Marks:1)

Vu-Topper RM

In Heapsort Algorithm, total time taken by heapify procedure is _____

A. $O(\log n)$ **Page 43**

B. $(\log^2 n)$

C. $(n \log n)$

D. $(n^2 \log n)$

Question No: 29

(Marks:1)

Vu-Topper RM

Al-Khwarizmi was a/an _____

Artist

A. Astronomer

B. Mathematication **Page 7**

C. Khalifah

Question No: 30

(Marks:1)

Vu-Topper RM

When matrix A of 5×3 is multiply with metric B of 3×4 then the number of multiplication required is: Not found exactly

A. 15

B. 12

C. 36

D. 60

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No: 31

(Marks:1)

Vu-Topper RM

Pseudo code of algorithms are to be read by _____.

A. People

Page 12

B. RAM

C. Computer

D. Compiler

Question No: 32

(Marks:1)

Vu-Topper RM

The sieve technique is a special case, where the number of sub-problems is Just _____

A. P

Page 34

B. 2

C. 3

D. 4

Question No: 33

(Marks:1)

Vu-Topper RM

When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has _____ sub-problems.

A. Overlapping

Google

B. Over costing

C. Optimized

D. Three

Question No: 34

(Marks:1)

Vu-Topper RM

In order to say anything meaningful about our algorithms, it will be important for us to settle on a _____.

A. Java Program

B. C++ Program

C. Pseudo program

D. Mathematically model of computation

Question No: 35

(Marks:1)

Vu-Topper RM

Merge sort is based on _____.

A. Brute-force

B. Plan-sweep

C. Axis-sweep

D. Divide and Conquer

Question No: 36

(Marks:1)

Vu-Topper RM

What time does Merge Sort algorithm take in order to sort an array of 'n' numbers?

A. (n)

B. (log n)

C. (n²)

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. $(n \log n)$

Question No: 37 (Marks:1)

Vu-Topper RM

algorithm, the first step is to _____.

A. Call Build-Heap procedure

Page 46

B. Sort the array in descending order

C. Call Heapify procedure

D. Find the number of input elements

Question No: 38 (Marks:1)

Vu-Topper RM

The definition of theta-notation relies on proving _____ asymptotic bound.

A. One

B. Lower

C. Upper

D. Both lower & upper

Page 25

Question No: 39 (Marks:1)

Vu-Topper RM

In merge sort algorithm, to merge two lists of size $n/2$ to a list of size n , takes _____ time.

A. Theta (n)

Page 32

B. Theta $\log(n)$

C. Theta $\log^2(n)$

D. Theta $n \log(n)$

Question No: 40 (Marks:1)

Vu-Topper RM

We can make _____ recursive calls in Fibonacci Sequence.

A. Infinite

B. Finite

Google

C. Only one

D. Zero

Question No: 41 (Marks:1)

Vu-Topper RM

Following is NOT the application of Edit Distance problem.

A. Speech recognition

B. Spelling Correction

C. Ascending Sort

Page 76

D. Computational Molecular Biology

Question No: 42 (Marks:1)

Vu-Topper RM

In plane sweep approach, a vertical line is swept across the 2d-plane and structure is used for holding the maximal points lying to the left of the sweep line.

A. Tree

B. Array

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

C. Queue

D. Stack

Question No: 43

(Marks:1)

Vu-Topper RM

Time will vary according to the nature of input data.

_____ time is the maximum running time over all legal inputs.

A. Worst-case

Page 13

B. Average-case

C. Best-case

D. Good-case

Question No: 44

(Marks:1)

Vu-Topper RM

Efficient algorithm requires less computational...

A. Memory

B. Running Time

C. Memory and Running Time

Page 9

D. Energy

Question No: 45

(Marks:1)

Vu-Topper RM

Selection algorithm takes theta _____

A. (n^2)

B. (n)

C. $\log(n)$

D. $n \log(n)$

Question No: 46

(Marks:1)

Vu-Topper RM

Time complexity of Dynamic Programming based algorithm for computing the minimum cost of Chain Matrix Multiplication is _____

A. $\log n$

B. n

C. n^2 (n square)

D. n^3 (n cube)

Page 90

Question No: 47

(Marks:1)

Vu-Topper RM

The Iteration method is used for _____

A. Solving Recurrence relations

Page 31

B. Merging elements in Merge sort

C. Comparing sorting algorithms only

D. Dividing elements in Merge sort

Question No: 48

(Marks:1)

Vu-Topper RM

In 3-Dimensional space, a point P has _____ coordinate(s).

A. (X, Y)

B. (X, 0)

C. (0, Y)

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. (X,Y, Z)

Question No: 49

(Marks:1)

Vu-Topper RM

Chain matrix multiplication problem can be solved through _____ strategy.

A. Dynamic programming

Page 85

B. Greedy

C. Divide and conquer

D. Sorting

Question No: 50

(Marks:1)

Vu-Topper RM

Merge sort have running time....running time of Heap sort. Not found exactly

A. Greater than

B. Less than

Google

C. Equal to

D. Different than

Question No: 51

(Marks:1)

Vu-Topper RM

The Omega-notation allows us to state only the asymptotic ___ bounds.

A. Middle

B. Lower

Page 25

C. Upper

D. Both lower & upper

Question No: 52

(Marks:1)

Vu-Topper RM

Both lower & upper Sorting can be in _____

A. Random order

B. Increasing order only

C. Decreasing order only

D. Both Increasing and Decreasing order

Question No: 53

(Marks:1)

Vu-Topper RM

Quicksort is a/an _____ and _____ sorting algorithm.

A. Not in place, not stable one

B. In place , not stable one

Page 54

C. In place , stable one

D. Not in place , stable one

Question No: 54

(Marks:1)

Vu-Topper RM

Consider three matrices X,Y,Z of dimensions 1×2 , 2×3 , 3×4 respectively. The number of multiplications of (XY) Z is:

A. 18

B. 32

C. 24

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. 30

Question No: 55

(Marks:1)

Vu-Topper RM

In Dynamic Programming, our approach is to _____

A. Express the problem non-recursively

B. Build the solution in a bottom-up fashion

Page 75

C. Develop the solution in a top-down fashion

D. Input several sub-problems simultaneously

Question No: 56

(Marks:1)

Vu-Topper RM

The knapsack problem is optimally solved by using brute force algorithm. Counting sort is suitable to sort the elements in range 1 to K;

A. K is large

B. K is small

Page 57

C. K may be large or small

D. None

Question No: 57

(Marks:1)

Vu-Topper RM

Matrix multiplication is a (n) _____ operation.

A. Commutative

B. Associative

Page 85

C. Neither commutative nor associative

D. Commutative but not associative

Question No: 58

(Marks:1)

Vu-Topper RM

In Dynamic Programming approach, solution is modified / changed

A. Always once

B. At each stage

Google

C. Only for specific problems

D. At 4th stage only

Question No: 59

(Marks:1)

Vu-Topper RM

In Knapsack problem, the goal is to put items in the Knapsack such that the value of the items is _____ subject to weight limit of knapsack.

A. Minimized

B. Decreased

C. Maximized

Page 91

D. None of the given options

Question No: 60

(Marks:1)

Vu-Topper RM

An in-place sorting algorithm is one that _____ uses additional array for storage.

A. Always

B. Permanently

C. Does not

Page 54

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. Sometime

Question No: 61

(Marks:1)

Vu-Topper RM

Dynamic Programming is a problem-solving approach in which____

- A. Problem is solved in Zero time
- B. Solution is developed only at final stage
- C. Both are correct

D. Both are incorrect

Google

Question No: 62

(Marks:1)

Vu-Topper RM

In Fibonacci sequence, each term is calculated by____ previous__ terms.

- A. Subtracting, Two
- B. Adding, Three

C. Adding, Two

Page 73

D. Multiplying, Two

Question No: 63

(Marks:1)

Vu-Topper RM

Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in an

- A. Class
- B. Array

C. Table

D. Variable

Question No: 64

(Marks:1)

Vu-Topper RM

Sorting is performed on the basis of _____.

- A. Computational resources
- B. Asymptotic notation
- C. Summation

D. Some key value of attribute

Page 39

Question No: 65

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm, we call Build-heap procedure _____.

- A. Twice
- B. Thrice

C. Only once

Page 46

D. As many times as we need

Question No: 66

(Marks:1)

Vu-Topper RM

In the statement “output P[1].x, P[1].y”, the number of times elements of P are accessed is _____.

A. 1

B. 2

Page 14

C. 3

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D.4

Question No: 67

(Marks:1)

Vu-Topper RM

_____ provides us more accurate result when input values are not closer with each other

- A. Mode
- B. Mean
- C. Average

D. Median

Page 34

Question No: 68

(Marks:1)

Vu-Topper RM

The process of _____ ends when you are left with such tiny pieces remaining that it is trivial to solve them.

- A. Brute-force
- B. Plan-sweep
- C. Axis-sweep

D. Divide and Conquer

Question No: 69

(Marks:1)

Vu-Topper RM

Rank of an element can be defined as _____.

- A. One minus the number of elements that are smaller
- B. Two plus the number of elements that are greater

C. One plus the number of elements that are smaller **Page 34**

D. Two minus the number of elements that are smaller

Question No: 70

(Marks:1)

Vu-Topper RM

If the time complexity of an algorithm is given by $O(1)$, then its time complexity would be

- A. Polynomial
- B. Exponential

C. Constant

Google

D. Average

Question No: 71

(Marks:1)

Vu-Topper RM

The asymptotic growth of $n(n+1)/2$ is:

- A. $O(n)$
- B. $O(n^2)$**
- C. $O(n+2)$
- D. $O(n \log n)$

Question No: 72

(Marks:1)

Vu-Topper RM

Approach of solving geometric problems by sweeping a line across the plane is called _____ sweep.

A. Line

B. Plane

Page 18

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- C. Cube
- D. Box

Question No: 73 (Marks:1) **Vu-Topper RM**

In Sieve technique, we solve the problem

A. In recursive manner **Page 34**

- B. Non recursively
- C. Using Merge Sort algorithm
- D. Using Brute force technique

Question No: 74 (Marks:1) **Vu-Topper RM**

One of the limitation in 0/1 knapsack is that an item can either be _____ in the bag or not.

A. Use

B. Put **Page 91**

- C. Move
- D. Store

Question No: 75 (Marks:1) **Vu-Topper RM**

Which one is not passed as parameter in Quick sort algorithm?

A. End of the array

B. Start of the array

C. Middle of the array

D. Array (containing input elements) **Google**

Question No: 76 (Marks:1) **Vu-Topper RM**

In the analysis of Selection algorithm, we get the convergent _____

A. Harmonic

B. Linear

C. Arithmetic

D. Geometric **Page 37**

Question No: 77 (Marks:1) **Vu-Topper RM**

A Random Access Machine (RAM) is an idealized machine with random access memory.

A. Infinite large **Page 10**

B. 512 MB

C. 256 MB

D. 2 GBs

Question No: 78 (Marks:1) **Vu-Topper RM**

While analyzing Selection algorithm, we make a number of passes, in fact it could be as many as

A. $n(n+1)$

B. $\log(n)$ **Page 37**

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- C. n/3
- D. n/4

Question No: 79 (Marks:1) **Vu-Topper RM**

In Random Access Machine (RAM), instructions are executed in

- A. Parallel
- B. Batch

- C. One by One**
- D. Multiple times

Page 10

Question No: 80 (Marks:1) **Vu-Topper RM**

In selection problem, the rank of an element will be its _____ position

- A. First
- C. Second last
- D. Last

B. final

Page 34

Question No: 81 (Marks:1) **Vu-Topper RM**

The worst-case running time of Merge sort is _____ in order to sort an array of n elements.

- A. $O(\log n)$
- B. $O(n)$

- C. $O(n \log n)$**
- D. $O(n)$

Page 40

Question No: 82 (Marks:1) **Vu-Topper RM**

$f(n)$ and $g(n)$ are asymptotically equivalent. This means that they have essentially the same _____.

- A. Size
- B. Results
- C. Variables

D. Growth Rates

Question No: 83 (Marks:1) **Vu-Topper RM**

An algorithm is a mathematical entity. Which is independent of _____.

- A. Programming language
- B. Machine and Programming language
- C. Compiler and Programming language

D. Programing Language Compiler and Machine

Question No: 84 (Marks:1) **Vu-Topper RM**

In Quick sort algorithm, Pivots form _____

- A. Stack
- B. Queue
- C. Graph

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. Binary Search Tree

Question No: 85

(Marks:1)

Vu-Topper RM

Counting sort is suitable for sorting the elements within range 1 to P.
where

A. P is large

B. P is Small

C. P is very large

D. P is undetermined

Question No: 86

(Marks:1)

Vu-Topper RM

In asymptotical analysis of $n^{(5/2)-3}$, as n becomes large, the dominant (fastest growing) term is some constant times

A. n_{-1}

B. n

C. n+1

D. $n * n^{p-23}$

Question No: 87

(Marks:1)

Vu-Topper RM

___ Items are not allowed in the 0/1 knapsack.

A. Lighter

B. Whole

C. Weighty

D. Fractional

Question No: 88

(Marks:1)

Vu-Topper RM

In partition algorithm, the subarray _____ has elements which are greater than pivot element x.

A. A[q]

B. A[p...r]

C. A[p...q-1]

D. S[q+1...r]

Question No: 89

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm, if heap property is violated

A. We ignore

B. We call Heapify procedure

C. We call Build Heap procedure

D. Heap property can never be violated

Question No: 90

(Marks:1)

Vu-Topper RM

_____ is not a characteristic of Random Access Machine.

A. Assigning a value to a variable

B. Locality of reference

C. Single-Processor

Page 10

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

D. Executing an arithmetic instruction

Question No: 91

(Marks:1)

Vu-Topper RM

The only way to convert an empty string into a string of j characters is by doing j insertions, represented as _____

A. $E(i,j) = 1$

B. $E(I,0) = I$

C. $E(0,j) = j$

Page 78

D. $E(1,j)= j$

Question No: 92

(Marks:1)

Vu-Topper RM

In Selection problem, the Sieve technique works in _____.

A. Non-recursive manner

B. Constant time

C. Phases

Page 34

D. One complete go

Question No: 93

(Marks:1)

Vu-Topper RM

Algorithm is a sequence of computational steps that _____ the input into output.

A. Merge

B. Assign

C. Transform

Page 7

D. Integrate

Question No: 94

(Marks:1)

Vu-Topper RM

If p_j dominates p_i and p_i dominates p_h then p_j also dominates p_h , it means dominance relation is

A. Transitive

Page 18

B. Non Transitive

C. Equation

D. Symbolic

Question No: 95

(Marks:1)

Vu-Topper RM

To find maximal points in brute-force algorithm each point of the space is compared against _____ of that space.

A. One other point

B. All other points

Page 11

C. Few other points

D. Most of the other points

Question No: 96

(Marks:1)

Vu-Topper RM

In the following code the statement "`cout<<j;`" executes _____ times.

`for (j=1; j<=5; j = j+2)`

`cout<<j;`

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- A. 5 times
- B. 2 times
- C. 3 times**
- D. 0 times

Question No: 97 (Marks:1) **Vu-Topper RM**
In merge sort algorithm, we split the array around the _____ index q.

A. Mid

Page 17

- B. Exiting
- C. Entring
- D. Summing

Question No: 98 (Marks:1) **Vu-Topper RM**
In Selection problem, the Sieve technique _____.

- A. Add some more input items each time
- B. Do not work recursively
- C. Do not uses Divide and Conquer approach

D. Eliminates undesired data items each time

Question No: 99 (Marks:1) **Vu-Topper RM**
Consider three matrices X, Y, Z of dimensions 1 x 2, 2 x 3, 3 x 4 respectively. The number of multiplications of X(YZ) is .

- A. 16
- B. 32**
- C. 32
- D. 26

Page 84

Question No:100 (Marks:1) **Vu-Topper RM**
In Heap Sort algorithm, the total running time for Heapify procedure is_

- A. Theta (log n)**
- B. Order (log n)
- C. Omega (log n)
- D. O(1) i.e. Constant time

Question No:101 (Marks:1) **Vu-Topper RM**
The sieve technique works where we have to find _____ items(s) from a large input.

A. Single

Page 34

- B. Two
- C. Three
- D. Similar

Question No:102 (Marks:1) **Vu-Topper RM**
In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object i , then we gain_____

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

A. W(Total Weight of Knapsack)

B. V (Total Value of all items)

C. vi (Value of object i)

Page 93

D. None of the given option

Question No:103

(Marks:1)

Vu-Topper RM

While Sorting, the order domain means for any two input elements x and y__ satisfies only.

A. $x < y$

Page 39

B. $x > y$

C. $x = y$

D. All of the above

Question No:104

(Marks:1)

Vu-Topper RM

For solving Selection problem, we introduced Sieve technique due to__

A. Using Decrease and Conquer strategy

Page 34

B. Avoiding to sort all input data

C. Eliminating Rank of an element

D. Using Brute-force approach

Question No:105

(Marks:1)

Vu-Topper RM

For solving Selection problem, we introduced Sieve technique due to__

A. Using Decrease and Conquer strategy

Page 34

B. Avoiding to sort all input data

C. Eliminating Rank of an element

D. Using Brute-force approach

Question No:106

(Marks:1)

Vu-Topper RM

In plane sweep approach, a vertical line is swept across the 2d-plane from_____.

A. Right to Left

B. Left to Right

Page 18

C. Top to Bottom

D. Bottom to top

Question No:107

(Marks:1)

Vu-Topper RM

For _____ values of n, any algorithm is fast enough.

A. Medium

B. Large

C. Small

Page 14

D. Infinity

Question No:108

(Marks:1)

Vu-Topper RM

Dynamic programming comprises of _____.

A. Recursion only

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- B. Repetition only
- C. Recursion with Repetition
- D. No Repetition but Recursion**

Page 75

Question No:109 (Marks:1) **Vu-Topper RM**

The function $f(n)=n(\log n+1)/2$ is asymptotically equal to $n \log n$:
Here Lower Bound means function $f(n)$ grows asymptotically at ___ as fast as $n \log n$.

- A. Least** **Page 23**
- B. Normal
- C. Most
- D. AT

Question No:110 (Marks:1) **Vu-Topper RM**

Counting sort has time complexity.

- A. $O(n+k)$
- B. $O(n)$** **Page 58**
- C. $O(k)$
- D. $O(n \log n)$

Question No:111 (Marks:1) **Vu-Topper RM**

Due to left complete nature of binary tree, the heap can be stored in

- A. Array** **Page 40**
- B. Structures
- C. Link List
- D. Stack

Question No:112 (Marks:1) **Vu-Topper RM**

Single item from a larger set of _____.

- A. Constant
- B. Pointers
- C. Phases
- D. n items** **Page 34**

Question No:113 (Marks:1) **Vu-Topper RM**

In the clique cover problem, for two vertices to be in the same group, they must be _____ each other.

- A. Apart from
- B. Far from
- C. Near to
- D. Adjacent to** **Page 76**

Question No:114 (Marks:1) **Vu-Topper RM**

How much time merge sort takes for an array of numbers?

- A. $T(n^2)$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- B. $T(n)$
- C. $T(\log n)$
- D. $T(n \log n)$**

Page 40

Question No:115 (Marks:1) **Vu-Topper RM**

In in-place sorting algorithm is one that uses arrays for storage.

- A. No additional array** **Page 54**
- B. An additional array
- C. Both of above may be true according to algorithm
- D. More than 3 arrays of one dimension

Question No:116 (Marks:1) **Vu-Topper RM**

Brute-force algorithm for 2D-Maxima is operated by comparing _____ pairs of points.

- A. Two
- B. Some
- C. Most
- D. All** **Page 18**

Question No:117 (Marks:1) **Vu-Topper RM**

While Sorting, the ordered domain means for any two input elements x and y _____ satisfies only.

- A. $x > y$
- B. $x < y$
- C. $x = y$
- D. All of the above** **Page 38**

Question No:118 (Marks:1) **Vu-Topper RM**

Quick sort is.

- A. Not stable but in place** **Page 54**
- B. Stable but not in place
- C. Stable & in Place
- D. Some time stable & some times in place

Question No:119 (Marks:1) **Vu-Topper RM**

Which may be a stable sort?

- A. Merger
- B. Insertion
- C. Both above** **Page 54**
- D. None of the above

Question No:120 (Marks:1) **Vu-Topper RM**

For the Sieve Technique we take time.

- A. $T(nk)$** **Page 34**
- B. $IT(n / 3)$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- C. n^2
- D. $n/$

Question No:121 (Marks:1) **Vu-Topper RM**

Continuation sort is suitable to sort the elements in range 1 to k.

- A. K is Large
- B. K is not known
- C. K may be small or large

D. K is small **Page 54**

Question No:122 (Marks:1) **Vu-Topper RM**

Asymptotic growth rate of the function is taken over _____ case running time. .

A. Worst

Page 14

- B. Average
- C. Best
- D. Normal

Question No:123 (Marks:1) **Vu-Topper RM**

Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their _____ coordinates. .

- A. Y
- B. Z
- C. X**
- D. X , Y

Question No:124 (Marks:1) **Vu-Topper RM**

In Quick sort, we don't have the control over the sizes of recursive calls.

- A. True** **Page 49**
- B. False
- C. Less information to decide
- D. Ether true or false

Question No:125 (Marks:1) **Vu-Topper RM**

Random access machine or RAM is a/an.

- A. Machine build by Al-Khwarizmi
- B. Mechanical machine
- C. Mathematical model** **Page 10**
- D. Electronics machine

Question No:126 (Marks:1) **Vu-Topper RM**

A heap is a left-complete binary tree that confirms to the _____.

- A. increasing order only
- B. decreasing order only

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:127

(Marks:1)

Vu-Topper RM

Which one of the following sorting algorithms is the fastest?

A. Merge sort

B. Quick sort

C. Insertion sort

D. Heap sort

Question No:128

(Marks:1)

Vu-Topper RM

Quick sort algorithm divide the entire array into _____ sub arrays.

A. 2

B. 3

C. 4

D. 5

Question No:129

(Marks:1)

Vu-Topper RM

In brute force algorithm, we measure running time $T(n)$ based on _____.

A. Worst-case time and best-case time

B. Worst-case time and average-case time

Page 46

C. Average-case time and best-case time

D. Best-case time and starting-case time

Question No:130

(Marks:1)

Vu-Topper RM

algorithm first of all _____.

A. Sorts all points

B. Delete some points

C. Output the elements

D. Pushes all points on stack

Question No:131

(Marks:1)

Vu-Topper RM

Which symbol is used for Omega notation?

A. (O)

B. (Θ)

C. (Ω)

D. (@)

Question No:132

(Marks:1)

Vu-Topper RM

Selection sort is a _____ sorting algorithm

A. In-place

Page 54

B. Not In-Place

C. Stable

D. in-partition

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:133

(Marks:1)

Vu-Topper RM

We do not need to prove comparison-based sorting algorithms by mathematically. It always takes _____ time.

A. Big Oh $n \log(n)$

B. Omega $n \log(n)$

C. Omega $n(n^2)$

D. Theta $n \log(n)$

Question No:134

(Marks:1)

Vu-Topper RM

Merge sort is a/an _____ and _____ sorting algorithm

A. Not in-place, not stable one

B. In-place, not stable one

C. In-place, stable one

D. Not in-place, stable one

Page 54

Question No:135

(Marks:1)

Vu-Topper RM

Cubic function will _____ a quadratic function.

A. Prove

B. Be equal to

C. Overtake

Page 25

D. Find

Question No:136

(Marks:1)

Vu-Topper RM

Insertion sort is a _____ sorting algorithm

A. Unstable

B. In-place

Page 54

C. Not In-Place

D. in-partition

Question No:137

(Marks:1)

Vu-Topper RM

To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is _____.

A. Big-oh notation

B. Theta notation

C. Omega notation

D. All of the given

Question No:138

(Marks:1)

Vu-Topper RM

Asymptotic growth of $8n^2 + 2n - 3$ is:

A. $\Theta(n^2 + n)$

B. $\Theta(n^2)$

Page 14

C. $\Theta(8n^2)$

D. $\Theta(8n^2 + 2n)$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:139

(Marks:1)

Vu-Topper RM

In the analysis of algorithms, _____ plays an important role.

A. Time

B. Money

C. Growth rate

D. Text analysis

Question No:140

(Marks:1)

Vu-Topper RM

In inductive approach of knapsack problem, we consider 2 cases, _____
Or _____.

A. Median, Mode

B. Recursive, Iterative

C. Leave object, Take object

Page 93

D. Sequentially. Parallel

Question No:141

(Marks:1)

Vu-Topper RM

Random Access Machine (RAM) can execute _____ instructions

A. Parallel

B. Only logical

C. Only arithmetic

D. Logical and arithmetic

Question No:142

(Marks:1)

Vu-Topper RM

Using _____ algorithm, efficiency is not given much importance

A. Greedy

B. Merge sort

C. Processing

D. Brute Force

Question No:143

(Marks:1)

Vu-Topper RM

Bubble sort takes theta _____ in the worst case

A. (n²)

Page 39

B. (n)

C. log(n)

D. nlog(n)

Question No:144

(Marks:1)

Vu-Topper RM

If matrix A of dimension $p \times q$ is multiply with matrix B of dimension $q \times r$, then dimension of resultant matrix is:

A. $q \times r$

B. $r \times p$

C. $P \times r$

D. $P \times q$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:145

(Marks:1)

Vu-Topper RM

Dynamic Programming algorithms often use some kind of _____ to store the results of intermediate sub-problems

- A. Stack
- B. Loop
- C. Table**
- D. variable

Question No:146

(Marks:1)

Vu-Topper RM

_____ is in-place sorting algorithm.

- A. Bubble sort** **Page 54**
- B. Merge sort
- C. Linear search
- D. Binary Search

Question No:147

(Marks:1)

Vu-Topper RM

Which one of the following problems can be solved using dynamic problem?

- A. Bubble sort problem
- B. Greedy search problem
- C. Fractional knapsack problem
- D. Matrix chain multiplication problem** **Page 85**

Question No:148

(Marks:1)

Vu-Topper RM

In chain matrix multiplication, solutions of the sub-problems are stored in a _____.

- A. Array
- B. Table** **Page 86**
- C. Tree
- D. Link list

Question No:149

(Marks:1)

Vu-Topper RM

What is the average running time of a quick sort algorithm?

- A. $O(n^2)$
- B. $O(n)$
- C. $O(n \log n)$** **Page 49**
- D. $O(\log n)$

Question No:150

(Marks:1)

Vu-Topper RM

Sorting Algorithms having O _____ running time are considered to be slow ones.

- A. (n)
- B. (n^2)** **Page 39**
- C. $(n \log(n))$
- D. $(\log(n))$

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:151

(Marks:1)

Vu-Topper RM

While solving Selection problem, in Sieve technique we partition input data_____

- A. Randomly
- B. According to Pivot**
- C. In increasing order
- D. In decreasing order

Question No:152

(Marks:1)

Vu-Topper RM

_____ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.

- A. Loop
- B. Function
- C. Recursion

D. Memoization

Page 74

Question No:153

(Marks:1)

Vu-Topper RM

In average-case time the probability of seeing input is denoted by ____.

- A. $p\{I\}$
- B. $p[I]$
- C. $p\langle i \rangle$

D. $p(i)$

Page 13

Question No:154

(Marks:1)

Vu-Topper RM

While applying the Sieve technique to selection sort, how to choose a pivot element.

- A. Through mean
- B. Linear

C. Randomly

Page 35

D. Sequentially

Question No:155

(Marks:1)

Vu-Topper RM

Number of _____ of the pseudo code are counted to measure the running time.

- A. Inputs
- B. Outputs

C. Steps

Page 13

D. Pages

Question No:156

(Marks:1)

Vu-Topper RM

Developing a dynamic programming algorithm generally involves _____ separate steps.

A. One

B. Two

Page 75

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- C. Three
- D. Four

Question No:157 (Marks:1) **Vu-Topper RM**
 $8n^2+2n+3$ will exceed $c28(n)$, no matter how large we make _____.

- A. n
- B. $2n$

C. c2 **Page 25**

- D. this quadratic equation

Question No:158 (Marks:1) **Vu-Topper RM**
The running time of quick sort algorithm_____.

- A. Is impossible to compute
- B. Has nothing to do with pivot selection
- C. Is Random upon each execution

D. Greatly influenced by the selection of pivot **Page 49**

Question No:159 (Marks:1) **Vu-Topper RM**
_____ involves breaking up the problem into sub problems whose solutions can be combined to solve the global problem.

- A. Complexity Theory
- B. Greedy Algorithms

C. Divide and Conquer Strategy **Page 34**

- D. Dynamic programming solution

Question No:160 (Marks:1) **Vu-Topper RM**
In _____ we have to find rank of an element from given input.

- A. Merge sort algorithm

B. Selection problem **Page 34**

- C. Brute force technique
- D. Plane Sweep algorithm

Question No:161 (Marks:1) **Vu-Topper RM**
How many steps are involved to design the dynamic programming strategy?

- A. 2

B. 4 **Page 92**

- C. 3
- D. 1

Question No:162 (Marks:1) **Vu-Topper RM**
In Bucket sort, if there are duplicates then each bin can be replaced by a Stack

- A. Heap
- B. Hash table

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:163

(Marks:1)

Vu-Topper RM

In merge sort algorithm, we split the array _____ to find index q.

- A. from end
- B. from start

C. midway

Page 28

- D. both from start or end

Question No:164

(Marks:1)

Vu-Topper RM

Find the maximum value of the items which can carry using knapsack

Knapsack weight capacity = 50.

Item Weight Value

11070

22020

33080

470 200

- A. 90
- B. 280
- C. 200**
- D. 100

Question No:165

(Marks:1)

Vu-Topper RM

In 2-d maxima problem a point p is said to be dominated by point q if_.

- A. $p.x \leq q.x$
- B. $p.x \leq q.x$ and $p.y \leq q.y$**
- C. $p.y \leq q.y$
- D. $p.x \geq q.x$ and $p.y \geq q.y$

Page 17

Question No:166

(Marks:1)

Vu-Topper RM

Sorting can be in _____.

- A. Increasing order only
- B. Decreasing order only
- C. Both increasing and decreasing order**
- D. Random order

Question No:167

(Marks:1)

Vu-Topper RM

Recurrence can be described in terms of _____.

- A. Array
- B. Linear
- C. Tree**
- D. Graph

Page 31

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:168

(Marks:1)

Vu-Topper RM

The brute-force algorithm for 2D-Maxima runs in order $O(_)$ time.

- A. n
- B. $n(\log n)$
- C. $n*n$**
- D. n^3

Page 18

Question No:169

(Marks:1)

Vu-Topper RM

In plane sweep approach of solving geometric problems, a _____ is swept across the plane.

- A. Line**
- B. Plane
- C. Cube
- D. Box

Page 18

Question No:170

(Marks:1)

Vu-Topper RM

Which of the following is calculated with Big Omega notation?

- A. Upper bounds
- B. Lower bounds**
- C. Medium bounds

Page 25

Question No:171

(Marks:1)

Vu-Topper RM

_____ is always based on divide and conquer strategy.

- A. Bubble sort
- B. Selection sort
- C. Pigeon sort
- D. Quick sort**

Page 46

Question No:172

(Marks:1)

Vu-Topper RM

If a matrix has three rows and two columns, then dimensions of matrix will be:

- A. 3×2**
- B. 2×3
- C. 3×3
- D. 2×2

Question No:173

(Marks:1)

Vu-Topper RM

Asymptotic notations are used to describe _____ of an algorithm.

- A. Size
- B. Length
- C. Running time**
- D. Compile time

Google

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:174

(Marks:1)

Vu-Topper RM

Catalan numbers are related the number of different ___ on 'n' nodes.

- A. Arrays
- B. linked lists
- C. binary trees**
- D. functions

Page 85

Question No:175

(Marks:1)

Vu-Topper RM

Applying the sieve technique to selection problem, _____ element is picked from array.

A. Pivot

Page 35

- B. Total
- C. Input
- D. Output

Question No:176

(Marks:1)

Vu-Topper RM

In recursive formulation of knapsack Problem: $V [0, j] = \underline{\hspace{2cm}}$ for $j \geq 0$

- A. 2
- B. -1**
- C. 1
- D. 2

Page 93

Question No:177

(Marks:1)

Vu-Topper RM

_____ is a linear time sorting algorithm.

- A. Merge sort
- B. Radix sort**
- C. Quick sort
- D. Bubble sort

Page 71

Question No:178

(Marks:1)

Vu-Topper RM

Quick sort is one of the _____ sorting algorithm.

- A. Fastest**
- B. Slowest
- C. Major
- D. Average

Page 19

Question No:179

(Marks:1)

Vu-Topper RM

The time assumed for each basic operation to execute on RAM model of computation is _____.

- A. Infinite
- B. Continuous
- C. Constant**
- D. Variable

Page 10

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:180 (Marks:1) **Vu-Topper RM**
While analyzing algorithms, _____ and _____ usually considered difficult to calculate.

- A. Floor, ceiling** **Google**
B. Row, Column
C. Finite, Infinite
D. Graph, Tree

Question No:181 (Marks:1) **Vu-Topper RM**
While analysis of the brute-force maxima algorithm, an array sorted in the reverse order is the type of _____ case input.

- A. Best
B. Worst **Page 14**
C. Somewhat bad
D. Average

Question No:182 (Marks:1) **Vu-Topper RM**
_____ is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

- A. Mean
B. Mode
C. Average
D. Median **Page 34**

Question No:183 (Marks:1) **Vu-Topper RM**
In asymptotical analysis of $n(n-3)$ and $4n*n$, as n becomes large, the dominant (fastest growing) term is some constant times _____.

- A. $n+1$
B. $n-1$
C. n
D. $n*n$ **Page 23**

Question No:184 (Marks:1) **Vu-Topper RM**
In addition to passing in the array itself to Merge Sort algorithm, we will pass in other arguments which are indices.

- A. Three
B. Two
C. Four
D. Five

Question No:185 (Marks:1) **Vu-Topper RM**
In 2d-maximal problem, a point is said to be if it is not dominated by any other point in that space.

- A. Member
B. Minimal

بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

C. Maximal

D. Joint

Question No:186

(Marks:1)

Vu-Topper RM

Counting sort assumes that the numbers to be sorted are in the range__.

A. K to n where n is large

B. K to n where k is small

C. 1 to k where k is small

D. k to n where n is small

Question No:187

(Marks:1)

Vu-Topper RM

Insertion sort is an efficient algorithm for sorting a _____ number of elements

A. Small

B. Large

C. Extra large

D. Medium

Question No:188

(Marks:1)

Vu-Topper RM

If the indices passed to merge sort algorithm are _____ then this means that there is only one element to sort.

A. Small

B. Large

C. Equal

D. Not Equal

Page 28

Question No:189

(Marks:1)

Vu-Topper RM

In Knapsack Problem, each item must be entirely accepted or rejected, is called _____ problem.

A. Linear

B. Fractional

C. 0-1

D. Optimal

Question No:190

(Marks:1)

Vu-Topper RM

If the time complexity of an algorithm is $O(n)$. then it is called _____ time complexity.

A. Linear

B. Constant

C. Average

D. Exponential

Question No:191

(Marks:1)

Vu-Topper RM

In the case of _____ analysis does not depend upon on the distribution of input.

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

- A. Merge sort
- B. Insertion sort
- C. Quick Sort**
- D. Heap sort

Question No:192 (Marks:1) **Vu-Topper RM**
We can use the _____ Property to devise a recursive formulation of the edit distance problem.

- A. Small substructure
- B. Algorithmic
- C. Real
- D. Optimal substructure**

Page 78

Question No:193 (Marks:1) **Vu-Topper RM**
The following sequence is called _____
1,2,3,5,8,13,21,34,55,.....

- A. Fibonacci sequence**
- B. Optimal sequence
- C. Optimize Sequence
- D. Overlapping sequence

Page 73

Question No:194 (Marks:1) **Vu-Topper RM**
Which one sorting algorithm is best suited to sort an array of 2 million elements?

- A. Insert sort
- B. Ridx Sort**
- C. Merge sort
- D. Quick sort

Page 71

Question No:195 (Marks:1) **Vu-Topper RM**
We can improve the performance of quick sort if we could be able to_.

- A. Select two or more pivots**
- B. Skip any sub-array completely
- C. Skit Input elements somehow
- D. Eliminate recursive calls

Page 34

Question No:196 (Marks:1) **Vu-Topper RM**
The problem with the brute-force algorithm is that it uses _____ in pruning out de

- A. Worst-case time
- B. No intelligence**
- C. Outside looping
- D. Artificial intelligence

Page 18

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:197

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm. Heapify procedure is _____ in nature.

A. Recursive

B. Non-Recursive

Page 43

C. Fast

D. Slow

Question No:198

(Marks:1)

Vu-Topper RM

An algorithm is said to be correct if for every _____ instance, it halts with the correct _____.

A. Input, Output

Page 13

B. Design, Analysis

C. Value, Key

D. Key, Analysis

Question No:199

(Marks:1)

Vu-Topper RM

If we have an equation $8n^2 + 7f \cdot n + 5f + 6$ then is large, _____ term will be much larger than the n term and will dominate the running time.

A. $f \cdot g(n)$

B. $g(n) * 2$

C. $n * 2$

Page 23

D. $f(n)$

Question No:200

(Marks:1)

Vu-Topper RM

For quick sort algorithm. Partitioning takes theta _____.

A. (n)

B. $\log(n)$

C. $n \log(n)$

D. $n^2 \log(n)$

Question No:201

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm, the maximum levels an element can move upward is _____.

A. Theta ($\log n$)

Page 43

B. Big-oh ($\log n$)

C. Omega ($\log n$)

D. $O(1)$ i.e. Constant time

Question No:202

(Marks:1)

Vu-Topper RM

Which process is used for avoiding unnecessary repetitions and looking them up again if we need them later.

A. Greedy Approach

B. Memoization

Page 74

C. Divide and conquer

D. Recursion

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:203

(Marks:1)

Vu-Topper RM

The worst-case running time of Quick sort is _____ in order to sort an array of n element.

A. $O(n \log n)$

Page 49

B. $O(n)$

C. $O(n^2)$

D. $O(\log n)$

Question No:204

(Marks:1)

Vu-Topper RM

Boolean operation is a _____ operation on an idealized RAM model of computation.

A. Advance

B. String

C. Basic

D. Normal

Question No:205

(Marks:1)

Vu-Topper RM

In chain matrix multiplication, if there are n items, there are _____ ways in which outer most pair of parentheses can placed.

A. n^2

B. $2n$

C. $n+1$

D. $n-1$

Page 85

Question No:206

(Marks:1)

Vu-Topper RM

The number of nodes in a complete binary tree of height h is:

A. $(h+1) - 1$

B. $(h+1)$

C. $2^{(h+1)} - 1$

Page 40

D. $((h+1)^2) - 1$

Question No:207

(Marks:1)

Vu-Topper RM

In Sieve Technique, we know the item of interest.

A. True

B. False

Question No:208

(Marks:1)

Vu-Topper RM

The Huffman codes provide a method of encoding data inefficiently when coded using ASCII standard.

A. True

B. False

Page 99

پری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

Question No:209

(Marks:1)

Vu-Topper RM

In Heap Sort algorithm, we build _____ for ascending sort.

A. Min heap

B. Max Heap

Page 41

C. Both

D. None of these

Question No:210

(Marks:1)

Vu-Topper RM

Quick sort is a recursive algorithm.

A. True

B. False

Vu-Topper RM



بری صحبت سے تنہائی بہتر ہے اور تنہائی سے نیک صحبت بہتر ہے

For More Help Contact What's app 03224021365

AL-JUNAID TECH INSTITUTE



CS502- FUNDAMENTALS OF
ALGORITHMS
(SOLVED MCQs)
FROM MIDTERM PAPERS
LECTURE (1-22)



BC190202640@vu.edu.pk
Junaidfazal08@gmail.com

For More Visit: vulmshelp.com

JUNAID MALIK
(0304-1659294)

▶▶▶

AL-JUNAID TECH INSTITUTE

**PAID SERVICE
CS619 PROJECTS**

Available training courses

- HTML
- JQUERY
- PHPMYSQL
- JAVASCRIPT
- BOOTSTRAPS
- NODEJS
- REACTJS
- CSS

LMS HANDLING

**PAID
ASSIGNMENTS, QUIZ & QDB**

**95% RESULTS
ALL LMS ACTIVITIES**

Contact Us :
+92 304 1659294
www.vulmshelp.com
junaidfazal08@gmail.com

AL-JUNAID TECH INSTITUTE

1. The sequence of merge sort algorithm is:
 - a. Divide Combine-Conquer
 - b. Conquer-Divide-Combine
 - c. Divide-Conquer-Combine** Page 27
 - d. Combine-Divide-Conquer
2. In _____ Knapsack Problem, limitation is that an item can either be put in the bag or not. Fractional items are not allowed.
 - a. 0
 - b. 1
 - c. 0/1** Page 91
 - d. Fractional
3. In Selection algorithm, we assume pivot selection takes theta _____ running time.
 - a. n** Page - 36
 - b. n^2
 - c. n^3
 - d. $\log(n)$
4. In Heap Sort algorithm (using max heap), when every time maximum elements removed from top _____.
 - a. We call merge Sort Algorithm
 - b. it becomes Order n^2 Algorithm
 - c. Divide and Conquer strategy helps us
 - d. We are left with a hole** Page - 41
5. If matrix A of dimension $p \times q$ is multiply with matrix B of dimension $q \times r$, then each entry in resultant matrix takes _____ time.
 - a. $O(q)$** Page - 84
 - b. $O(1)$
 - c. $O(p \times q)$
 - d. $O(q \times r)$

AL-JUNAID TECH INSTITUTE

6. _____ is a method of solving a problem in which we check all possible solutions to the problem to find the solution we need.
- Plane-Sweep Algorithm
 - Sorting Algorithm
 - Brute-Force Algorithm** google
 - Greedy approach
7. The worst case running time of Quick sort algorithm _____.
- Cannot be quadratic
 - Is quadratic**
 - Is always Exponential
 - Is linear
8. In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with _____ leaf in the tree.
- second last
 - Last** Page -41
 - First
 - Any
9. Quick sort algorithm was developed by -
- AlferdAho
 - Sedgewick
 - John Vincent Atanasoff
 - Tony Hoare** – Google wikipedia
10. If Matrix-A has dimensions "3x2" and Matrix-B has dimensions "2x3", then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.
- 3x2
 - 2x3
 - 2x2
 - 3x3** <http://www.calcul.com/show/calculator/matrix-multiplication>
11. For comparison-based sorting algorithms, it is possible to sort more efficiently than Omega $n \log(n)$ time.
- Always
 - Not** P-54
 - Sometimes

AL-JUNAID TECH INSTITUTE

d. Sometimes not

12. Dynamic Programming approach is usually useful in solving optimization problems.

a. True

b. False

13. In Sorting the key value or attribute _____ from an ordered domain.

a. Must be page 39

b. Not always

c. May be

d. Occasionally

14. Result of asymptotical analysis of $n(n-3)$ and $4n*n$ is that _____

a. $n(n-1)$ is asymptotically Less

b. $n(n-1)$ is asymptotically Greater

c. Both are asymptotically Not equivalent

d. Both are asymptotically Equivalent page 23 ($4n*n = 4n^2$)

15. Floor and ceiling are _____ to calculate while analyzing algorithms a. Very easy

b. Usually considered difficult P-31

c. 3rd Option is missing

d. 4th Option is missing

16. _____ of reference is an important fact of current processor technology.

a. Defining

b. Assigning

c. Formality

d. Locality P-8

17. In max-heap, largest element is stored at root node. Where is the smallest element stored?

a. Right Node

b. Leaf Node

c. Middle Node

d. Left Node

AL-JUNAID TECH INSTITUTE

18. In average-case time analysis of Quick sort algorithm, the most balanced case for partition is when we divide the list of elements into _.

a. Equal no. of pieces as of input elements

- b. Single piece exactly
- c. Two nearly equal pieces
- d. Three nearly equal pieces

19. Which of the following is calculated with Big O notation?

a. Medium bounds

b. Upper bounds Page - 25

- c. Lower bounds
- d. Both upper and lower bounds

20. Edit distance algorithm based on _____ strategy

a. Greedy

b. Dynamic Programming Page - 81

- c. Divide and Conquer
- d. Searching

21. In Heapsort Algorithm, total time taken by heapify procedure is _____

a. $O(\log n)$ Page-43

- b. $O(\log^2 n)$
- c. $O(n \log n)$
- d. $O(n^2 \log n)$

22. Al-Khwarizmi was a/an _____

a. Artist

b. Mathematician P-7

- c. Astronomer
- d. Khalifah

23. When matrix A of 5×3 is multiply with metric B of 3×4 then the number of multiplication required is: Not found exactly

a. 15

b. 12

c. 36

d. 60 Not Found exactly but as per formula at page 84,

AL-JUNAID TECH INSTITUTE

24. Pseudo code of algorithms are to be read by _____.

- a. People Page -12
- b. RAM
- c. Computer
- d. Compiler

25. The sieve technique is a special case, where the number of sub-problems is Just

- _____
- a. 1 P-34
 - b. 2
 - c. 3
 - d. 4

26. When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has _____ sub-problems.

- a. Overlapping - Google Search
- b. Over costing
- c. Optimized
- d. Three

27. Sieve technique is very important special case of Divide-and-Conquer strategy.

- a. True P-34
- b. False

28. In order to say anything meaningful about our algorithms, it will be important for us to settle on a _____.

- a. Java Program
- b. C++ Program
- c. Pseudo program
- d. Mathematical model of computation P-10

29. Merge sort is based on _____.

- a. Brute-force
- b. Plan-sweep
- c. Axis-sweep
- d. Divide and Conquer P-27

AL-JUNAID TECH INSTITUTE

30. What time does Merge Sort algorithm take in order to sort an array of 'n' numbers?

- a. (n)
- b. (log n)
- c. (n²)

d. (n log n) Google Search

31. In Heap Sort algorithm, the first step is to _____.

a. Call Build-Heap procedure Page - 46

- b. Sort the array in descending order
- c. Call Heapify procedure
- d. Find the number of input elements

32. The definition of theta-notation relies on proving _____ asymptotic bound.

- a. One
- b. Lower
- c. Upper

d. Both lower & upper Page - 25

33. In merge sort algorithm, to merge two lists of size n/2 to a list of size n, takes _____ time.

a. Theta (n) Page - 32

- b. Theta log(n)
- c. Theta log²(n)
- d. Theta n log(n)

34. We can make _____ recursive calls in Fibonacci Sequence.

a. Infinite

b. Finite google

- c. Only one
- d. Zero

35. Following is NOT the application of Edit Distance problem.

- a. Speech recognition
- b. Spelling Correction

c. Ascending Sort Page - 76

d. Computational Molecular Biology

AL-JUNAID TECH INSTITUTE

36. In plane sweep approach, a vertical line is swept across the 2d-plane and structure is used for holding the maximal points lying to the left of the sweep line.

- a. Array
- b. Queue
- c. Stack**
- d. Tree

Page - 18

37. When a heapify procedure is applied to the root node to restore the heap, then at each level, the comparison performed takes time:

- a. It will take $(\log n)$.
- b. It can not be predicted
- c. It will take $O(1)$.**
- d. Time will vary according to the nature of input data.

Page - 43

38. _____ time is the maximum running time over all legal inputs.

- a. Worst-case**
- b. Average-case
- c. Best-case
- d. Good-case

Page - 13

39. Efficient algorithm requires less computational...

- a. Memory
- b. Running Time
- c. Memory and Running Time**
- d. Energy

Page - 9

40. For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from _____

- a. half of the input values
- b. all possible random values**
- c. Pivot is input separately
- d. values greater than 5

Page - 50

41. Selection algorithm takes theta _____

- a. (n^2)
- b. (n)**
- c. $\log(n)$
- d. $n \log(n)$

Page - 37

AL-JUNAID TECH INSTITUTE

42. Recurrence can be described in terms of a tree.

a. Yes Page - 31

b. No

43. Time complexity of Dynamic Programming based algorithm for computing the minimum cost of Chain Matrix Multiplication is _____

a. Log n

b. n

c. n^2 (n square)

d. n^3 (n cube) Page -90

44. The Iteration method is used for _____

a. Comparing sorting algorithms only

b. Solving Recurrence relations Page 31

c. Merging elements in Merge sort

d. Dividing elements in Merge sort

45. In 3-Dimensional space, a point P has _____ coordinate(s).

a. (X, Y)

b. (X, 0)

c. (0, Y)

d. (X, Y, Z)

46. Chain matrix multiplication problem can be solved through _____ strategy.

a. Dynamic programming Page - 85

b. Greedy

c. Divide and conquer

d. Sorting

47. Merge sort have running time....running time of Heap sort. Not found exactly

a. Greater than

b. Less than Google

c. Equal to

d. Different than

48. Median is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

a. True

b. False Page - 34

AL-JUNAID TECH INSTITUTE

49. We do not need to mathematically prove that for comparison-based sorting algorithms always takes Omega $n \log(n)$ time.
- a. True Google & VU Tech (pg 46 not very clear)
 - b. False
50. The Omega-notation allows us to state only the asymptotic _____ bounds.
- a. Middle
 - b. Lower Page 25
 - c. Upper
 - d. Both lower & upper
51. Both lower & upper Sorting can be in _____
- a. Increasing order only
 - b. Decreasing order only
 - c. Both Increasing and Decreasing order GOOGLR Search
 - d. Random order
52. Radix sort performs sorting the numbers _____ digit (s) at a time.
- a. One Page - 71
 - b. Two
 - c. Three
 - d. All
53. Quicksort is a/an _____ and _____ sorting algorithm.
- a. Not in place, not stable one
 - b. In place , not stable one Page - 54
 - c. In place , stable one
 - d. Not in place , stable one
54. Consider three matrices X,Y,Z of dimensions $1 \times 2, 2 \times 3, 3 \times 4$ respectively. The number of multiplications of (XY) Z is:
- a. 18 As per lecture slides
 - b. 32
 - c. 24
 - d. 30
55. In Fibonacci Sequence, unnecessary repetitions do not exist at all.
- a. True
 - b. False Page – 74

AL-JUNAID TECH INSTITUTE

56. It is not a Fibonacci sequence . 1,1,1,2,3,5,8,13,21,34,55,.....
- a. True Page - 73
 - b. False
57. Heap sort is a/ an _____ and _____ sorting algorithm.
- a. Not in place, not stable one
 - b. In place , not stable one Page - 54
 - c. In place , stable one
 - d. Not in place , stable one
58. Identify the True Statement
- a. The knapsack problem does not belong to the domain of optimization problems.
 - b. The knapsack problem belongs to the domain of optimization problems. Page - 91
 - c. The Knapsack problem cannot be solved by using dynamic programming
 - d. The knapsack problem is optimally solved by using brute force algorithm.
59. In Dynamic Programming, our approach is to _____
- a. Develop the solution in a top-down fashion
 - b. Express the problem non-recursively
 - c. Build the solution in a bottom-up fashion Page - 75
 - d. Input several sub-problems simultaneously
60. Counting sort is suitable to sort the elements in range 1 to K;
- a. K is large
 - b. K is small Page - 57
 - c. K may be large or small
 - d. None
61. We can multiply two matrices A and B only when they are compatible which means
- a. Number of columns in A must be equal to number of rows in B. it seems Correct as per page 84
 - b. Number of rows and columns do not matter

AL-JUNAID TECH INSTITUTE

- c. Number of columns in A must be equal to number of columns in B
- d. Number of rows in A must be equal to number of rows in B

62. Matrix multiplication is a (n) _____ operation.

- a. Commutative
- b. Associative** Page 85
- c. Neither commutative nor associative
- d. Commutative but not associative

63. In Dynamic Programming approach, solution is modified / changed

- a. Always once
- b. At each stage** google and wikipedia
- c. Only for specific problems
- d. At 4th stage only

64. In Knapsack problem, the goal is to put items in the Knapsack such that the value of the items is _____ subject to weight limit of knapsack.

- a. Minimized
- b. Decreased
- c. Maximized** Page - 91
- d. None of the given options

65. An in-place sorting algorithm is one that _____ uses additional array for storage.

- a. Always
- b. Permanently
- c. Does not** Page - 54
- d. Sometime

66. Memoization is a part of Dynamic Programming Strategy.

- a. True** Page - 74
- b. False

67. If matrix A of dimension 2×4 is multiply with matrix B of dimension 4×3 , then the dimension of resultant matrix is _____ Not found exactly

- a. 2×4
- b. 4×3
- c. 3×4
- d. 2×3** It seems correct as per second last Para of page 84

AL-JUNAID TECH INSTITUTE

68. In Dynamic Programming approach, we do not store the solution to each sub-problem in case if it reappears.
- a. True
 - b. False** Page - 75
69. Dynamic Programming is a problem-solving approach in which____
- a. Problem is solved in Zero time
 - b. Solution is developed only at final stage
 - c. Both are correct
 - d. Both are incorrect** google
70. In Fibonacci sequence, each term is calculated by____ previous__ terms.
- a. Subtracting, Two
 - b. Adding, Three
 - c. Adding, Two** Page - 73
 - d. Multiplying, Two
71. Selection sort is not an in-place sorting algorithm.
- a. True** Page - 54
 - b. False
72. If there are $\theta(n^2)$ entries in edit distance matrix then the total running time is:
- a. $\theta(n)$
 - b. $\theta(1)$
 - c. $\theta(n^2)$** Page - 84
 - d. $\theta(n \log n)$
73. The only way to convert a string of i characters into the empty string is with i deletions, represented as
- a. $E(0.j) = j$
 - b. $E(i.j) = 1$
 - c. $E(0.i) = j$
 - d. $E(i.0) = i$** Page - 78
74. Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in an
- a. Array
 - b. Table** Page - 86
 - c. Variable
 - d. class

AL-JUNAID TECH INSTITUTE

75. We can use the optimal substructure property to devise a formulation of the edit distance problem.
- a. Selective
 - b. Optimum
 - c. Iterative
 - d. Recursive Page - 78
76. Sorting is performed on the basis of _____.
- a. Computational resources
 - b. Asymptotic notation
 - c. Summation
 - d. Some key value of attribute page- 39
77. In Heap Sort algorithm, we call Build-heap procedure _____.
- a. Only once page 46
 - b. Twice
 - c. Thrice
 - d. As many times as we need
78. Radix sort is not a non-comparative integer sorting algorithm.
- a. True Google Search
 - b. False
79. In the statement “output P[1].x, P[1].y”, the number of times elements of P are accessed is _____.
- a. 1
 - b. 2 page 14
 - c. 3
 - d. 4
80. The main purpose of mathematical analysis is measuring the _____ required by the algorithm.
- a. Space
 - b. Execution time P-13
 - c. Inputs & outputs
 - d. Execution time and memory

AL-JUNAID TECH INSTITUTE

81. _____ provides us more accurate result when input values are not closer with each other

a. Average

b. Median P-34

c. Mode

d. Mean

82. The process of _____ ends when you are left with such tiny pieces remaining that it is trivial to solve them.

a. Brute-force

b. Plan-sweep

c. Divide and Conquer P-27

d. Axis-sweep

83. _____ overcomes the limitations of _____ by working as per positional notations of numbers.

a. Counting sort, Radix sort

b. Radix sort, Counting sort P-71

84. Memorization is a part of Dynamic Programming strategy.

a. True P-74

b. False

85. Rank of an element can be defined as _____.

a. One minus the number of elements that are smaller

b. Two plus the number of elements that are greater

c. One plus the number of elements that are smaller P-34

d. Two minus the number of elements that are smaller

86. If the time complexity of an algorithm is given by $O(1)$, then its time complexity would be

a. Polynomial

b. Exponential

c. Constant - Wikipedia

d. Average

87. Quick sort is a recursive algorithm.

a. True Wikipedia ; Google

b. False

AL-JUNAID TECH INSTITUTE

88. The asymptotic growth of $n(n+1)/2$ is:

- a. $O(n^2)$ As the n^2 term has the largest contribution, the Big-O complexity is $O(n^2)$
- b. $O(n)$
- c. $O(n+2)$
- d. $O(n \log n)$

89. Approach of solving geometric problems by sweeping a line across the plane is called _____ sweep.

- a. Line
- b. Plane Page 18
- c. Cube
- d. Box

90. As per algorithm of Dynamic Programming, we need to store

- a. First sub-problem only
- b. Best solution only
- c. Intermediate sub-problems Pg:75
- d. Final solution only

91. In Sieve technique, we solve the problem

- a. In recursive manner Pg:34
- b. Non recursively
- c. Using Merge Sort algorithm
- d. Using Brute force technique

92. One of the limitation in 0/1 knapsack is that an item can either be _____ in the bag or not.

- a. Use
- b. Put Pg:91
- c. Move
- d. Store

93. Which one is not passed as parameter in Quick sort algorithm?

- a. End of the array
- b. Middle of the array
- c. Array (containing input elements) Google
- d. Start of the array

AL-JUNAID TECH INSTITUTE

94. In the analysis of Selection algorithm, we get the convergent _____
- a. Harmonic
 - b. Linear
 - c. Arithmetic
 - d. Geometric Pg:37**
95. A Random Access Machine (RAM) is an idealized machine with random access memory.
- a. Infinite large Pg:10**
 - b. 512 MB
 - c. 256 MB
 - d. 2 GBs
96. While analyzing Selection algorithm, we make a number of passes, in fact it could be as many as
- a. $n(n+1)$
 - b. $\log(n)$ Pg:37**
 - c. $n/3$
 - d. $n/4$
97. In Random Access Machine (RAM), instructions are executed in
- a. Parallel
 - b. Batch
 - c. One by One Pg:10
 - d. Multiple times
98. In selection problem, the rank of an element will be its _____ position
- a. First
 - b. final Pg:34**
 - c. Second last
 - d. Last
99. The worst-case running time of Merge sort is _____ in order to sort an array of n elements.
- a. $O(\log n)$
 - b. $O(n)$
 - c. $O(n \log n)$ page 40 and google**
 - d. $O(n)$

AL-JUNAID TECH INSTITUTE

100. $f(n)$ and $g(n)$ are asymptotically equivalent. This means that they have essentially the same _____.
- a. Results
 - b. Variables
 - c. Size
 - d. Growth rates** P:23
101. An algorithm is a mathematical entity. Which is independent of _____.
- a. Programming language
 - b. Machine and Programming language
 - c. Compiler and Programming language
 - d. Programming language Compiler and Machine** P:07
102. In Quick sort algorithm, Pivots form _____
- a. Stack
 - b. Queue
 - c. Binary Search Tree P:49
 - d. Graph
103. Counting sort is suitable for sorting the elements within range 1 to P. where
- a. P is large
 - b. P is small** P-57
 - c. P is very large
 - d. P is undetermined
104. In asymptotical analysis of $n^{(5/2)-3}$, as n becomes large, the dominant (fastest growing) term is some constant times
- a. n_1
 - b. n
 - c. $n+1$
 - d. $n*n$ P-23
105. _____ Items are not allowed in the 0/1 knapsack.
- a. Lighter
 - b. Fractional** P-91
 - c. Whole
 - d. Weighty

AL-JUNAID TECH INSTITUTE

106. Fibonacci Sequence was named on _____, a famous mathematician in 12th Century.
- a. Fred Brooks
 - b. Grady Booch
 - c. Leonardo Pisano** **P-73**
 - d. Edgar F. Codd
107. In Heap Sort algorithm, we build _____ for ascending sort.
- a. Max heap** **P-41**
 - b. Min heap
108. Bubble sort is not an in-place sorting algorithm.
- a. True
 - b. False** **P-54**
109. In partition algorithm, the subarray _____ has elements which are greater than pivot element x.
- a. $A[p..r]$
 - b. $A[p..q-1]$
 - c. $A[q]$
 - d. $A[q+1..r]$** **P-46**
110. In Heap Sort algorithm, if heap property is violated
- a. We call Build heap procedure** **P-43**
 - b. We call Heapify procedure
 - c. We ignore
 - d. Heap property can never be violated
111. _____ is not a characteristic of Random Access Machine.
- a. Single-Processor **P-10**
 - b. Assigning a value to a variable
 - c. Locality of reference
 - d. Executing an arithmetic instruction
112. The only way to convert an empty string into a string of j characters is by doing j insertions, represented as _____
- a. $E(i,j) = 1$
 - b. $E(l,0) = l$
 - c. $E(0,j) = j$** **page 78**
 - d. $E(1,j) = j$

AL-JUNAID TECH INSTITUTE

113. In Selection problem, the Sieve technique works in _____.
- a. Non-recursive manner
 - b. Constant time
 - c. Phases** page 34
 - d. One complete go
114. Algorithm is a sequence of computational steps that ---- the input into output.
- a. Merge
 - b. Assign
 - c. Transform** page 7
 - d. Integrate
115. If p_j dominates p_i and p_i dominates p_h then p_j also dominates p_h , it means dominance relation is
- a. Transitive page 18
 - b. Non Transitive
 - c. Equation
 - d. Symbolic
116. To find maximal points in brute-force algorithm each point of the space is compared against _____ of that space.
- a. One other point
 - b. All other points** page 11
 - c. Few other points
 - d. Most of the other points
117. In the following code the statement "cout<<j;"executes ---- times. for (j=1; j<=5; j = j+2)
- ```
cout<<j;
```
- a. 5 times
  - b. 2 times
  - c. 3 times**
  - d. 0 times

# AL-JUNAID TECH INSTITUTE

118. In merge sort algorithm, we split the array around the \_\_\_\_\_ index q. a. Entering  
**b. Mid** page 17  
c. Exiting  
d. Summing
119. In Selection problem, the Sieve technique \_\_\_\_\_.  
a. Add some more input items each time  
b. Do not work recursively  
c. Do not uses Divide and Conquer approach  
**d. Eliminates undesired data items each time**
120. Consider three matrices X, Y, Z of dimensions 1 x 2, 2 x 3, 3 x 4 respectively. The number of multiplications of X(YZ) is .  
a. 16  
b. 32  
c. 26  
**d. 32** page 84
121. In Heap Sort algorithm, the total running time for Heapify procedure is \_\_\_\_\_  
**a. Theta (log n)**  
b. Order (log n)  
c. Omega (log n)  
d. O(1) i.e. Constant time
122. The sieve technique works where we have to find \_\_\_\_\_ items(s) from a large input.  
**a. Single** page 34  
b. Two  
c. Three  
d. Similar
123. In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object i , then we gain \_\_\_\_\_  
a. W(Total Weight of Knapsack)  
b. V (Total Value of all items)  
**c.  $v_i$  (Value of object i)** page 93  
d. None of the given option

# AL-JUNAID TECH INSTITUTE

124. While Sorting, the order domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.
- a.  $x < y$  page 39
  - b.  $x > y$
  - c.  $x = y$
  - d. All of the above
125. For solving Selection problem, we introduced Sieve technique due to \_\_\_\_\_
- a. Using Decrease and Conquer strategy page 34
  - b. Avoiding to sort all input data
  - c. Eliminating Rank of an element
  - d. Using Brute-force approach
126. \_\_\_\_\_ is one of the few problems, where provable lower bounds exist on how fast we can sort.
- a. Searching
  - b. Sorting page 38
  - c. Both Searching & sorting
  - d. Growing
127. In plane sweep approach, a vertical line is swept across the 2d-plane from \_\_\_\_\_.
- a. Right to Left
  - b. Left to Right page 18
  - c. Top to Bottom
  - d. Bottom to top
128. In generating Fibonacci sequence, we can avoid unnecessary repetitions by \_\_\_\_\_ process.
- a. Tokenization
  - b. Memorization page 43
  - c. Randomization
  - d. Memorization
129. For \_\_\_\_\_ values of  $n$ , any algorithm is fast enough.
- a. Small page 14
  - b. Medium
  - c. Large
  - d. Infinity

# AL-JUNAID TECH INSTITUTE

130. Dynamic programming comprises of \_\_\_\_\_.
- Recursion only
  - Repetition only
  - Recursion with Repetition
  - No Repetition but Recursion** page 75
131. The function  $f(n)=n(\log n+1)/2$  is asymptotically equal to  $n \log n$ : Here Lower Bound means function  $f(n)$  grows asymptotically at \_\_\_ as fast as  $n \log n$ .
- Least** page 23
  - Normal
  - Most
  - At
132. Counting sort has time complexity.
- $O(n+k)$
  - $O(n)$**  page 58
  - $O(k)$
  - $O(n \log n)$
133. Due to left complete nature of binary tree, the heap can be stored in
- Array** page 40
  - Structures
  - Link List
  - Stack
134. Single item from a larger set of \_\_\_\_\_.
- Constant
  - Pointers
  - Phases
  - n items** page 34
135. In the clique cover problem, for two vertices to be in the same group, they must be \_\_\_\_\_ each other.
- Apart from
  - Far from
  - Near to
  - Adjacent to** page 76

# AL-JUNAID TECH INSTITUTE

136. How much time merge sort takes for an array of numbers?
- a.  $T(n^2)$
  - b.  $T(n)$
  - c.  $T(\log n)$
  - d.  $T(n \log n)$  page 40**
137. In in-place sorting algorithm is one that uses arrays for storage.
- a. No additional array page 54**
  - b. An additional array
  - c. Both of above may be true according to algorithm
  - d. More than 3 arrays of one dimension
138. Brute-force algorithm for 2D-Maxima is operated by comparing \_\_\_\_\_ pairs of points.
- a. Two
  - b. Some
  - c. Most
  - d. All page 18**
139. While Sorting, the ordered domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.
- a.  $x > y$
  - b.  $x < y$
  - c.  $x = y$
  - d. All of the above page 38**
140. Quick sort is.
- a. Stable & in place
  - b. Not stable but in place page 54**
  - c. Stable but not in place
  - d. Some time stable & some times in place
141. Which may be a stable sort?
- a. Merger
  - b. Insertion
  - c. Both above page 54**
  - d. None of the above

# AL-JUNAID TECH INSTITUTE

142. For the Sieve Technique we take time.
- a.  $T(nk)$  page 34
  - b.  $IT(n / 3)$
  - c.  $n^2$
  - d.  $n/3$
143. Continuation sort is suitable to sort the elements in range 1 to k.
- a. K is Large
  - b. K is not known
  - c. K may be small or large
  - d. K is small page 54
144. Asymptotic growth rate of the function is taken over \_\_\_\_\_ case running time. .
- a. Best
  - b. Worst page 14
  - c. Average
  - d. Normal
145. The sieve technique is a special case, where the number of sub problems is just.
- a. 5
  - b. Many
  - c. 1 page 34
  - d. Few
146. In Quick sort, we don't have the control over the sizes of recursive calls.
- a. True page 49
  - b. False
  - c. Less information to decide
  - d. Either true or false
147. Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their \_\_\_\_\_ coordinates. .
- a. X page 18
  - b. Y
  - c. Z
  - d. X , Y

# AL-JUNAID TECH INSTITUTE

148. Random access machine or RAM is a/an.
- a. Machine build by Al-Khwarizmi
  - b. Mechanical machine
  - c. Mathematical model** page 10
  - d. Electronics machine
149. The Huffman codes provide a method of encoding data inefficiently when coded using ASCII standard.
- a. True
  - b. False page 99
150. A heap is a left-complete binary tree that confirms to the \_\_\_\_\_.
- a. increasing order only
  - b. decreasing order only
  - c. heap order** page 40
  - d. log n order
151. If we associate  $(x, y)$  integers pair to cars where  $x$  is the speed of the car and  $y$  is the negation of the price. High  $y$  value for a car means a \_\_\_\_\_ car.
- a. Fast
  - b. Slow
  - c. Expensive
  - d. Cheap**
152. Which one of the following sorting algorithms is the fastest?
- a. Merge sort
  - b. Quick sort**
  - c. Insertion sort
  - d. Heap sort
153. Quick sort algorithm divide the entire array into \_\_\_\_\_ sub arrays.
- a. 2**
  - b. 3
  - c. 4
  - d. 5
154. In brute force algorithm, we measure running time  $T(n)$  based on \_\_\_\_\_.
- a. Average-case time and best-case time
  - b. Worst-case time and average-case time** page 46
  - c. Worst-case time and best-case time
  - d. Best-case time and starring-case time

# AL-JUNAID TECH INSTITUTE

155. For 2D Maxima problem. Plane Sweep algorithm first of all \_\_\_\_\_.
- a. Sorts all points
  - b. Delete some points
  - c. Output the elements
  - d. Pushes all points on stack
156. There are \_\_\_\_\_ entries in the Edit Distance Matrix
- a.  $\Theta(n)$
  - b.  $\Theta(n^2)$  page 84
  - c.  $\Theta(n+2)$
  - d.  $\Theta(n + 100)$
157. Which symbol is used for Omega notation?
- a. (O)
  - b. (e)
  - c. ( $\Omega$ )
  - d. (@)
158. Selection sort is a \_\_\_\_\_ sorting algorithm
- a. In-place page 54
  - b. Not In-Place
  - c. Stable
  - d. in-partition
159. In Dynamic Programming based solution of knapsack problem, to compute entries of 'V', we will imply a(n) \_\_\_\_\_ approach.
- a. Subjective
  - b. Inductive
  - c. Brute Force
  - d. Combination
160. We do not need to prove comparison-based sorting algorithms by mathematically. It always takes \_\_\_\_\_ time.
- a. Big Oh  $n \log(n)$
  - b. Omega  $n \log(n)$  NOT SURE
  - c. Omega  $n(n^2)$
  - d. Theta  $n \log(n)$

# AL-JUNAID TECH INSTITUTE

161. Merge sort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm
- a. Not in-place, not stable one
  - b. In-place, not stable one
  - c. In-place, stable one
  - d. Not in-place, stable one** page 54
162. Cubic function will \_\_\_\_\_ a quadratic function.
- a. Prove
  - b. be equal to
  - c. overtake** Page 25
  - d. find
163. Insertion sort is a \_\_\_\_\_ sorting algorithm
- a. Unstable
  - b. In-place** Page 54
  - c. Not In-Place
  - d. in-partition
164. To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is \_\_\_\_\_.
- a. Big-oh notation
  - b. Theta notation**
  - c. Omega notation
  - d. All of the given
165. Asymptotic growth of  $8n^2 + 2n - 3$  is:
- a.  $\Theta(n^2 + n)$
  - b.  $\Theta(n^2)$**  page 14
  - c.  $\Theta(8n^2)$
  - d.  $\Theta(8n^2 + 2n)$
166. In the analysis of algorithms, \_\_\_\_\_ plays an important role.
- a. text analysis
  - b. time**
  - c. growth rate
  - d. money

# AL-JUNAID TECH INSTITUTE

167. In inductive approach of knapsack problem, we consider 2 cases, \_\_\_\_\_

Or \_\_\_\_\_.

a. Median, Mode

b. Recursive, Iterative

c. Leave object, Take object page 93

d. Sequentially. Parallel

168. Random Access Machine (RAM) can execute \_\_\_\_\_ instructions

a. only logical

b. parallel

c. only arithmetic

d. logical and arithmetic

169. Using \_\_\_\_\_ algorithm, efficiency is not given much importance

a. Greedy

b. Merge sort

c. Processing as there is no algorithm by this name

d. Brute Force

170. Bubble sort takes theta \_\_\_\_\_ in the worst case

a.  $(n^2)$  page 39

b.  $(n)$

c.  $\log(n)$

d.  $n\log(n)$

171. If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then dimension of resultant matrix is:

a.  $p \times q$

b.  $p \times r$  page 84

c.  $q \times r$

d.  $r \times p$

172. Dynamic Programing algorithms often use some kind of \_\_\_\_\_ to store the results of intermediate sub-problems

a. table (Page 75)

b. variable

c. stack

d. loop

# AL-JUNAID TECH INSTITUTE

173. \_\_\_\_\_ is in-place sorting algorithm.
- a. Bubble sort (Page 54)
  - b. Merge sort
  - c. Linear search
  - d. Binary Search
174. Which one of the following problems can be solved using dynamic problem?
- a. Bubble sort problem
  - b. Matrix chain multiplication problem page 85
  - c. Greedy search problem
  - d. Fractional knapsack problem
175. In chain matrix multiplication, solutions of the sub-problems are stored in a \_\_\_\_\_.
- a. Array
  - b. Table page 86
  - c. Tree
  - d. Link list
176. What is the average running time of a quick sort algorithm?
- a.  $O(n^2)$
  - b.  $O(n)$
  - c.  $O(n \log n)$  (Page 49)
  - d.  $O(\log n)$
177. Sorting Algorithms having  $O$  \_\_\_\_\_ running time are considered to be slow ones.
- a.  $(n)$
  - b.  $(n^2)$  (Page 39)
  - c.  $(n \log(n))$
  - d.  $(\log(n))$
178. While solving Selection problem, in Sieve technique we partition input data \_\_\_\_\_.
- a. In increasing order
  - b. In decreasing order
  - c. According to Pivot
  - d. Randomly

# AL-JUNAID TECH INSTITUTE

179. \_\_\_\_\_ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.
- a. Loop
  - b. Memoization page 74
  - c. Recursion
  - d. Function
180. In average-case time the probability of seeing input is denoted by \_\_\_\_\_.
- a.  $p\{I\}$
  - b.  $p[I]$
  - c.  $p\langle i \rangle$
  - d.  $p(i)$  page 13
181. While applying the Sieve technique to selection sort, how to choose a pivot element.
- a. Through mean
  - b. Linear
  - c. Randomly page 35
  - d. Sequentially
182. Number of \_\_\_\_\_ of the pseudo code are counted to measure the running time.
- a. Inputs
  - b. Outputs
  - c. Steps page 13
  - d. Pages
183. Developing a dynamic programming algorithm generally involves \_\_\_\_\_ separate steps.
- a. One
  - b. Two page 75
  - c. Three
  - d. Four
184.  $8n^2+2n+3$  will exceed  $c28(n)$ , no matter how large we make \_\_\_\_\_.
- a.  $n$
  - b.  $2n$
  - c.  $c2$  page 25
  - d. this quadratic equation

# AL-JUNAID TECH INSTITUTE

185. The running time of quick sort algorithm\_\_\_\_\_.
- a. Is impossible to compute
  - b. Has nothing to do with pivot selection
  - c. Is Random upon each execution
  - d. Greatly influenced by the selection of pivot** page 49
186. \_\_\_\_\_ involves breaking up the problem into sub problems whose solutions can be combined to solve the global problem.
- a. Complexity Theory
  - b. Dynamic programming solution
  - c. Divide and Conquer Strategy** page 34
  - d. Greedy Algorithms
187. In \_\_\_\_\_ we have to find rank of an element from given input.
- a. Merge sort algorithm
  - b. Selection problem** page 34
  - c. Brute force technique
  - d. Plane Sweep algorithm
188. How many steps are involved to design the dynamic programming strategy?
- a. 2
  - b. 3
  - c. 1
  - d. 4** page 92
189. In Bucket sort, if there are duplicates then each bin can be replaced by a
- a. Stack
  - b. Linked list** page 69
  - c. Hash table
  - d. Heap
190. In merge sort algorithm, we split the array \_\_\_\_\_ to find index q.
- a. from start
  - b. midway** page 28
  - c. from end
  - d. both from start or end

# AL-JUNAID TECH INSTITUTE

191. Find the maximum value of the items which can carry using knapsack Knapsack weight capacity = 50.

Item Weight Value

11070

22020

33080

470 200

a. 280

b. 100

c. 90

d. 200

192. In 2-d maxima problem a point p is said to be dominated by point q if

\_\_\_\_\_.

a.  $p.x \leq q.x$

b.  $p.x \leq q.x$  and  $p.y \leq q.y$  page 17

c.  $p.y \leq q.y$

d.  $p.x \geq q.x$  and  $p.y \geq q.y$

193. Sorting can be in \_\_\_\_\_.

a. Increasing order only

b. Decreasing order only

c. Both increasing and decreasing order

d. Random order

194. Recurrence can be described in terms of \_\_\_\_\_.

a. Array

b. Linear

c. Tree page 31

d. Graph

195. The brute-force algorithm for 2D-Maxima runs in order  $O(\_\_)$  time.

a. n

b.  $n(\log n)$

c.  $n*n$  page 18

d.  $n^3$

# AL-JUNAID TECH INSTITUTE

196. In plane sweep approach of solving geometric problems, a \_\_\_\_\_ is swept across the plane.
- a. Line page 18
  - b. Plane
  - c. Cube
  - d. Box
197. Which of the following is calculated with Big Omega notation?
- a. Medium bounds
  - b. Upper bounds
  - c. Lower bounds Page - 25
  - d. Both upper and lower bounds
198. \_\_\_\_\_ is always based on divide and conquer strategy.
- a. Bubble sort
  - b. Selection sort
  - c. Pigeon sort
  - d. Quick sort page 46
199. If a matrix has three rows and two columns, then dimensions of matrix will be:
- a. 3x2
  - b. 2x3
  - c. 3x3
  - d. 2x2
200. Asymptotic notations are used to describe \_\_\_\_\_ of an algorithm.
- a. Length
  - b. running time google
  - c. size
  - d. compile time
201. Catalan numbers are related the number of different \_\_\_\_\_ on 'n' nodes.
- a. Arrays
  - b. linked lists
  - c. binary trees page 85
  - d. functions

# AL-JUNAID TECH INSTITUTE

202. Applying the sieve technique to selection problem, \_\_\_\_\_ element is picked from array.
- a. Output
  - b. Total
  - c. Input
  - d. Pivot** page 35
203. Dynamic Programming approach is usually useful in solving \_\_\_\_\_ problems.
- a. Normal
  - b. Optimization** google
  - c. Array
  - d. Loop
204. In recursive formulation of knapsack Problem:  $V[0, j] = \underline{\hspace{2cm}}$  for  $j \geq 0$
- a. -1
  - b. 0** page 93
  - c. 1
  - d. 2
205. \_\_\_\_\_ is a linear time sorting algorithm.
- a. Merge sort
  - b. Radix sort** page 71
  - c. Quick sort
  - d. Bubble sort
206. Quick sort is one of the \_\_\_\_\_ sorting algorithm.
- a. Fastest** page 19
  - b. Slowest
  - c. Major
  - d. Average
207. The time assumed for each basic operation to execute on RAM model of computation is \_\_\_\_\_.
- a. Infinite
  - b. Continuous
  - c. Constant** page 10
  - d. Variable

# AL-JUNAID TECH INSTITUTE

208. In Sieve Technique, we know the item of interest. a. True  
b. False page 34
209. While analyzing algorithms, \_\_\_\_\_ and \_\_\_\_\_ usually considered difficult to calculate.  
a. Finite, Infinite  
b. Floor, ceiling google  
c. Row, Column  
d. Graph, Tree
210. While analysis of the brute-force maxima algorithm, an array sorted in the reverse order is the type of \_\_\_\_\_ case input.  
a. Best  
b. Worst page 14  
c. Somewhat bad  
d. Average
211. \_\_\_\_\_ is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.  
a. Mean  
b. Mode  
c. Average  
d. Median page 34
212. In asymptotical analysis of  $n(n-3)$  and  $4n*n$ , as  $n$  becomes large, the dominant (fastest growing) term is some constant times \_\_\_\_\_.  
a.  $n+1$   
b.  $n-1$   
c.  $n$   
d.  $n*n$  page 23
213. In addition to passing in the array itself to Merge Sort algorithm, we will pass in other arguments which are indices.  
a. Two P-38  
b. Three  
c. Four  
d. Five

# AL-JUNAID TECH INSTITUTE

214. In 2d-maximal problem, a point is said to be if it is not dominated by any other point in that space.
- a. Member
  - b. Minimal
  - c. Maximal** **P-11**
  - d. Joint
215. Counting sort assumes that the numbers to be sorted are in the range \_\_\_\_\_.
- a. K to n where n is large
  - b. 1 to k where k is small (P-57)
  - c. K to n where k is small
  - d. k to n where n is small
216. Insertion sort is an efficient algorithm for sorting a \_\_\_\_\_ number of elements
- a. Large** **P-39**
  - b. Small
  - c. Extra large
  - d. Medium
217. If the indices passed to merge sort algorithm are \_\_\_\_\_ then this means that there is only one element to sort.
- a. Small** page 28
  - b. Large
  - c. Equal
  - d. Not Equal
218. In Knapsack Problem, each item must be entirely accepted or rejected, is called \_\_\_\_\_ problem.
- a. Fractional
  - b. 0-1** **P-92**
  - c. Linear
  - d. Optimal

# AL-JUNAID TECH INSTITUTE

219. If the time complexity of an algorithm is  $O(n)$ . then it is called \_\_\_\_\_ time complexity.

**a. Linear** [Wikipedia](#)

b. Constant

c. Average

d. Exponential

220. In the case of \_\_\_\_\_ analysis does not depend upon on the distribution of input.

a. Merge sort

**b. Quick sort** [P-50](#)

c. Insertion sort

d. Heap sort

221. We can use the \_\_\_\_\_ Property to devise a recursive formulation of the edit distance problem.

a. Small substructure

b. Algorithmic

c. Real

**d. Optimal substructure** [page 78](#)

222. The following sequence is called \_\_\_\_\_

1,2,3,5,8,13,21,34,55,.....

a. Optimize sequence

**b. Fibonacci sequence** [page 73](#)

c. Optimal sequence

d. Overlapping sequence

223. Which one sorting algorithm is best suited to sort an array of 2 million elements?

a. Bubble sort

b. Insert sort

c. Merge sort

d. Quick sort

**e. Ridx Sort** [page 71](#)

# AL-JUNAID TECH INSTITUTE

224. We can improve the performance of quick sort if we could be able to \_\_,\_\_\_\_\_.
- a. Skip input elements somehow
  - b. Select two or more pivots** page 34
  - c. Skip any sub-array completely
  - d. Eliminate recursive calls
225. The problem with the brute-force algorithm is that it uses \_\_\_\_\_ in pruning out de
- a. Worst-case time
  - b. No intelligence** page 18
  - c. Outside looping
  - d. Artificial intelligence
226. In chain matrix multiplication, the order of the matrices \_\_\_\_\_.
- a. Can be changed
  - b. Can not be changed** page 85
  - c. is equal
  - d. is reverse
227. In quick sort algorithm, we choose pivot \_\_\_\_\_.
- a. Always the smallest element
  - b. Greater than 5
  - c. Randomly** page 35
  - d. Less than 5
228. In Heap Sort algorithm. Heapify procedure is \_\_\_\_\_ in nature.
- a. Recursive
  - b. Non-Recursive** page 43
  - c. Fast
  - d. Slow
229. When matrix A of  $5 \times 3$  is multiplied with matrix B of  $3 \times 4$  then the number of multiplications required will be \_\_\_\_\_.
- a. 15
  - b. 12
  - c. 36
  - d. 60**

# AL-JUNAID TECH INSTITUTE

230. An algorithm is said to be correct if for every \_\_\_\_\_ instance, it halts with the correct \_\_\_\_\_.

**a. Input, Output** page 13

b. Design, Analysis

c. Value, Key

d. Key, Analysis

231. In chain matrix multiplication, table is filled \_\_\_\_\_ to find the multiplication of matrix.

a. row wise

b. column wise

c. diagonally

**d. bottom-to-up** page 86

232. If we have an equation  $8n^2 + 7f \cdot n + 5f + 6$  then is large, \_\_\_\_\_ term will be much larger than the  $n$  term and will dominate the running time.

a.  $f \cdot g(n)$

b.  $g(n) \cdot 2$

**c.  $n \cdot 2$**  page 23

d.  $f(n)$

233. For quick sort algorithm. Partitioning takes theta \_\_\_\_\_.

a.  $(n)$

b.  $\log(n)$

c.  $n \log(n)$

**d.  $n^2 \log(n)$**

234. In Heap Sort algorithm, the maximum levels an element can move upward is \_\_\_\_\_.

**a. Theta ( $\log n$ )** page 43

b. Big-oh ( $\log n$ )

c. Omega ( $\log n$ )

d.  $O(1)$  i.e. Constant time

235. \_\_\_\_\_ programming is essentially recursion without repetition.

a. Fast

**b. Dynamic** page 75

c. Array

# AL-JUNAID TECH INSTITUTE

d.  $n (\log n)$

236. In heap sort algorithm, the total running time for heavily procedure is \_.

- Big-oh( $\log n$ )
- $O(1)$  i.e. Constant time
- Theta ( $\log n$ )
- Omega ( $\log n$ )

237. Quick sort algorithm is required a lot of comparison in the \_\_\_\_\_ condition.

- Worse case
- Best and average case
- Average case
- Best case

238. In heap sort algorithm (using max heap). When every time maximum element is removed from top.

- Divide and conquer strategy helps us
- We are left with a lot
- We call merge sort algorithm
- It becomes order  $n^2$  algorithm

239. In average-case time analysis of quick sort algorithm, The most balanced case for partition is where we divide the list of element into \_\_\_\_\_

- Three nearly equal pieces
- Single piece exactly
- Two nearly piece
- Equal no. of piece as of input element

240. Consider three matrices X,Y,Z dimensions  $1 \times 2.2 \times 3 \times 4$  respectively. The number of multiplication of (XYZ) is:

- 32

# AL-JUNAID TECH INSTITUTE

- 30
- 24

- 18

241. Quicksort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm.

- In-place, not stable one

- Not in-place, stable one
- In-place, stable one
- Not in-place, not stable one

242. \_\_\_\_\_ items are not allowed in the 0/1 knapack.

- Lighter
- Whole
- Weighty

- Fractional

243. The main purpose of mathematical analysis is measuring the \_\_\_\_\_ required by the algorithm.

- Space
- Execution time and memory

- Input & output
- Execution time

244. Execution time of an algorithm can be measured by \_\_\_\_\_.

- Divide and conquer approach
- Both brute force and divide and conquer approach

- Mathematical analysis

- Brute force approach

245. Quick sort is based on \_\_\_\_\_ strategy.

- Graph theory

- Divide-and-conquer

- Dynamic programming
- Greedy approach

# AL-JUNAID TECH INSTITUTE

246. A sorting algorithm is called as \_\_\_\_\_ if duplicate element remain in the same relative position after sorting.

- $O(n)$  algorithm
- Stable
- Parallel
- Complex

247. Which one sorting algorithm is best suited to sort an array of 2 million elements?

- Insert sort
- Quick sort
- Merge sort
- Bubble sort

248. We can use the \_\_\_\_\_ property to devise a recursive formulation of the edit distance problem.

- Algorithm
- Small substructure
- Optimal substructure
- Real

249. While sorting. The ordered domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.

- All of the above
- $x > y$
- $x < y$
- $x = y$

250.  $8n^2 + 2n - 3$  will eventually exceed  $c2*(n)$  no matter how large we make \_\_\_\_\_.

- $2n$
- $n$
- this equation

# AL-JUNAID TECH INSTITUTE

- c2

251. \_\_\_\_\_ is a method of solving a problem in which we check all possible solution to the problem to find the solution we need.

- Sorting algorithm
- Greedy approach
- Plane-sweep algorithm
- Brute-force algorithm

252. In quick sort algorithm, provost form \_\_\_\_\_.

- Graph
- Stack
- Binary search tree

- Queue

253. In asymptotical analysis of  $n(n-3)$  and  $4n*n$ , as  $n$  becomes large, the dominant (fastest growing) term is some constant time \_\_\_\_\_

- $n+1$
- $n*n$
- $n$
- $n-1$

254. If Matrix-A has dimensions "3×2" and Matrix-B has dimensions "2×3", then multiplication of Matrix -A and Matrix-B will result a new Matrix-C having dimensions

- 2×3
- 2×2
- 3×2
- 3×3

255. Boolean operation is a \_\_\_\_\_ operation on an idealized RAM model of computation.

- Advance
- Normal

# AL-JUNAID TECH INSTITUTE

- Basic

- Starting

256. There are \_\_\_\_\_ entries in the Edit Distance Matrix.

- $O(n^2)$

- $O(n+100)$

- $O(n)$

- $O(n+2)$

257. Counting sort is suitable for sorting the elements within range 1 to P. where \_\_\_\_\_

- P is undetermined

- P is small

- P is very large

- P is large

258. Suppose we have 4 matrices A,B,C,D. what is correct expansion of  $m[1,2]$  in chain matrix multiplication?

- $m[1,2] = m[1,1] + m[2,2] + p_0 \cdot p_1 \cdot p_3$

- $m[1,2] = m[1,1] + m[2,2] + p_0 \cdot p_1 \cdot p_3$

- $m[1,2] = m[1,1] + m[2,2] + p_0 \cdot p_1 \cdot p_3$

- $m[1,2] = m[1,1] + m[2,2] + p_0 \cdot p_1 \cdot p_3$

259. Which one is not passed as parameter in Quick Sort algorithm?

- Array (containing input elements)

- Middle of the array

- Start of the array

- End of the array

260. In asymptotical analysis of  $n^{(5+2)}-3$ . As n becomes large, the dominant (fastest growing) term is some constant times \_\_\_\_\_

- $n+1$

- $n*n$

- n

# AL-JUNAID TECH INSTITUTE

- $n-1$

261. For \_\_\_\_\_ values of  $n$ , any algorithm is fast enough.

- Medium

- **Small**

- Infinity

- Large

262. Dynamic programming algorithms often use some kind of \_\_\_\_\_ to store the result of intermediate sub-problems.

- Stack

- Loop

- **Table**

- Variable

263. In selection problem, the Sieve technique works in \_\_\_\_\_

- One complete go

- Constant time

- Non-recursive manner

- **Phases**

264. In heap Sort algorithm, the maximum levels an element can move upward is \_\_\_\_\_.

- **Theta ( $\log n$ )**

- $O(1)$  i.e. Constant time

- Omega ( $\log n$ )

- Big-oh ( $\log n$ )

265. While analysis of the brute-force maxima algorithm, an array started in the reverse order is the type of \_\_\_\_\_ case input.

- **Worst**

- Best

- Somewhat

- Average

# AL-JUNAID TECH INSTITUTE

266. What type of instructions Random Access Machine (RAM) can execute?

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ Arithmetic and logic (Page 10)
- ▶ Parallel and recursive

267. For Chain Matrix Multiplication we can not use divide and conquer approach because,

- ▶ We do not know the optimum k (Page 86)
- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time
- ▶ Size of data is not given

268. What is the total time to heapify?

- ▶  $O(\log n)$  (Page 43)
- ▶  $O(n \log n)$
- ▶  $O(n^2 \log n)$
- ▶  $O(\log^2 n)$

269. \_\_\_\_\_ is a graphical representation of an algorithm

- ▶ notation
- ▶ notation
- ▶ Flowchart
- ▶ Asymptotic notation

270. A RAM is an idealized machine with \_\_\_\_\_ random-access memory.

- ▶ 256MB
- ▶ 512MB
- ▶ an infinitely large (Page 10)
- ▶ 100GB

271. What type of instructions Random Access Machine (RAM) can execute? Choose best answer

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ Arithmetic and logic
- ▶ Parallel and recursive

272. What is the solution to the recurrence  $T(n) = T(n/2) + n$ .

- ▶  $O(\log n)$
- ▶  $O(n)$  (Page 37)
- ▶  $O(n \log n)$
- ▶  $O(n^2)$

Mid

**Quiz by Attiq Kundi**

**SUBSCRIBE “[Knowledge For Us](#)” ON YOU TUBE**

**100 % CORRECT MCSQ**

**AND Many MoRe**

**CS-502**

**MIDTERM EXAMINATION**

**Fall 2011**

**CS502- Fundamentals of Algorithms**

**Question No: 1 ( Marks: 1 ) - Please choose one**

Due to left complete nature of binary tree, the heap can be stored in

- ▶ **Arrays (Page 40)**
- ▶ Structures
- ▶ Link Lis
- ▶ Stack

**Question No: 1 ( Marks: 1 ) - Please choose one**

What type of instructions Random Access Machine (RAM) can execute?

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic

▶ **Arithmetic and logic (Page 10)**

- ▶ Parallel and recursive

**Question No: 1 ( Marks: 1 ) - Please choose one**

For Chain Matrix Multiplication we can not use divide and conquer approach because,

- ▶ **We do not know the optimum k (Page 86)**
- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time
- ▶ Size of data is not given

**Question No: 1 ( Marks: 1 ) - Please choose one**

What is the total time to heapify?

- ▶  **$O(\log n)$  (Page 43)**
- ▶  $O(n \log n)$
- ▶  $O(n^2 \log n)$
- ▶  $O(\log^2 n)$

**Question No: 1 ( Marks: 1 ) - Please choose one**

word Algorithm comes from the name of the muslim author \_\_\_\_\_

- ▶ **Abu Ja'far Mohammad ibn Musa al-Khowarizmi.**

**Question No: 1 ( Marks: 1 ) - Please choose one** al-Khwarizmi's work was written in a book titled \_\_\_\_\_

- ▶ **al Kitab al-mukhatasar fi hisab al-jabr wa'l-muqabalah**

## MIDTERM EXAMINATION

Spring 2010

### CS502- Fundamentals of Algorithms

**Question No: 1 ( Marks: 1 ) - Please choose one**

Random access machine or RAM is a/an

- ▶ Machine build by Al-Khwarizmi
- ▶ Mechanical machine
- ▶ Electronics machine
- ▶ **Mathematical model (Page 10)**

**Question No: 2 ( Marks: 1 ) - Please choose one**

\_\_\_\_\_ is a graphical representation of an algorithm

- ▶  $\Sigma$  notation
- ▶  $\Theta$  notation
- ▶ **Flowchart [Click here for detail](#)**
- ▶ Asymptotic notation

**Question No: 3 ( Marks: 1 ) - Please choose one**

A RAM is an idealized machine with \_\_\_\_\_ random-access memory.

- ▶ 256MB
- ▶ 512MB
- ▶ **an infinitely large (Page 10)**
- ▶ 100GB

**Question No: 4 ( Marks: 1 ) - Please choose one**

What type of instructions Random Access Machine (RAM) can execute? Choose best answer

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ **Arithmetic and logic (Rep)**
- ▶ Parallel and recursive

**Question No: 5 ( Marks: 1 ) - Please choose one**

What will be the total number of max comparisons if we run brute-force maxima algorithm with n elements?

- ▶  $n^2$
- ▶  $n$
- ▶  $n^2$
- ▶  $n$  (Page 14)
- ▶  $n^8$

**Question No: 6 ( Marks: 1 ) - Please choose one**

What is the solution to the recurrence  $T(n) = T(n/2) + n$ .

- ▶  $O(\log n)$
- ▶  $O(n)$  (Page 37)
- ▶  $O(n \log n)$
- ▶  $O(n^2)$

**Question No: 7 ( Marks: 1 ) - Please choose one**

Consider the following code:

```

For(j=1; j<n; j++)
 For(k=1; k<15; k++)
 For(l=5; l<n; l++)
 {

```

```
Do_something_constant();
```

```
}
```

What is the order of execution for this code.

- ▶  $O(n^3)$
- ▶  $O(n^2)$
- ▶  $O(n^2 \log n)$
- ▶  $O(n^2)$

**Question No: 8 ( Marks: 1 ) - Please choose one**

What is the total time to heapify?

- ▶  $O(\log n)$  rep
- ▶  $O(n \log n)$
- ▶  $O(n^2 \log n)$
- ▶  $O(\log^2 n)$

**Question No: 9 ( Marks: 1 ) - Please choose one**

Consider the following Algorithm:

```
Factorial (n){
```

```
 if (n=1)
```

```
 return 1
```

```
 else
```

```
 return (n * Factorial(n-1))
```

**Recurrence for the following algorithm is:**

- ▶  $T(n) = T(n-1) + 1$
- ▶  $T(n) = nT(n-1) + 1$
- ▶  $T(n) = T(n-1) + n$
- ▶  $T(n) = T(n(n-1)) + 1$

**Question No: 10 ( Marks: 1 ) - Please choose one**

When we call heapify then at each level the comparison performed takes time

▶ **It will take  $\Theta(1)$  (Page 43)**

- ▶ Time will vary according to the nature of input data
- ▶ It can not be predicted
- ▶ It will take  $\Theta(\log n)$

**Question No: 11 ( Marks: 1 ) - Please choose one**

In Quick sort, we don't have the control over the sizes of recursive calls

▶ **True(Page 40)**

- ▶ False
- ▶ Less information to decide
- ▶ Either true or false

**Question No: 12 ( Marks: 1 ) - Please choose one** Is it possible to sort without making comparisons?

▶ **Yes(Page 57)**

- ▶ No

**Question No: 13 ( Marks: 1 ) - Please choose one**

If there are  $\Theta(n^2)$  entries in edit distance matrix then the total running time is ▶  $\Theta(1)$

▶  **$\Theta(n^2)$  [Click here for detail](#)**

- ▶  $\Theta(n)$
- ▶  $\Theta(n \log n)$

**Question No: 14 ( Marks: 1 ) - Please choose one**

For Chain Matrix Multiplication we can not use divide and conquer approach because,

▶ **We do not know the optimum k (Page 86)**

- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time

- ▶ Size of data is not given

**Question No: 15 ( Marks: 1 ) - Please choose one**

The Knapsack problem belongs to the domain of \_\_\_\_\_ problems.

▶ **Optimization (Page 91)**

- ▶ NP Complete
- ▶ Linear Solution
- ▶ Sorting

**Question No: 16 ( Marks: 1 ) - Please choose one**

Suppose we have three items as shown in the following table, and suppose the capacity of the knapsack is 50 i.e.  $W = 50$ .

| Item | Value | Weight |
|------|-------|--------|
| 1    | 60    | 10     |
| 2    | 100   | 20     |
| 3    | 120   | 30     |

The optimal solution is to pick

- ▶ Items 1 and 2
- ▶ Items 1 and 3
- ▶ **Items 2 and 3 (correct)**
- ▶ None of these

**MIDTERM EXAMINATION**  
**Spring 2010**  
**CS502- Fundamentals of Algorithms**

**Question No: 1 ( Marks: 1 ) - Please choose one**

For the Sieve Technique we take time

- ▶ **T(nk) (Page 34)**
- ▶ T(n / 3)
- ▶ n<sup>2</sup>
- ▶ n/3

**Question No: 1 ( Marks: 1 ) - Please choose one**

Sieve Technique applies to problems where we are interested in finding a single item

from a larger set of \_\_\_\_\_

Select correct option:

- ▶ **n items (Page 34)**
- ▶ phases
- ▶ pointers
- ▶ constant

**Question No: 1 ( Marks: 1 ) - Please choose one**

\_\_\_\_\_ graphical representation of algorithm.

- ▶ asymptotic

► **Flowchart (rep)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

who invented the quick sort

► **C.A.R. Hoare** [Click here for detail](#)

**Question No: 1 ( Marks: 1 ) - Please choose one**

main elements to a divide-and-conquer

► **Divide, conquer, combine (Page 27)**

**Question No: 1 ( Marks: 1 ) - Please choose one** Mergesort  
is a stable algorithm but not an in-place algorithm.

► **True (Page 54)**

► false

**Question No: 1 ( Marks: 1 ) - Please choose one**

Counting sort the numbers to be sorted are in the range 1 to k where k is small.

► **True (Page 57)**

► False

**MIDTERM EXAMINATION**

**Spring 2007**

**CS502- Fundamentals of Algorithms**

**Question No: 1 ( Marks: 1 ) - Please choose one**

Total time for heapify is:

$2$

▶  $O(\log n)$

▶  $O(n \log n)$

$2$

▶  $O(n \log n)$

▶  **$O(\log n)$**

**Rep**

**Question No: 1 ( Marks: 1 ) - Please choose one**

If an algorithm has a complexity of  $\log_2 n + n \log_2 n + n$ . we could say that it has complexity

▶  $O(n)$

▶  $O(n \log_2 n)$

▶  $O(3)$

▶  $O(\log_2 (\log_2 n))$

▶  $O(\log_2 n)$

**Question No: 1 ( Marks: 1 ) - Please choose one** In RAM model instructions are executed

▶ **One after another (Page 10)**

▶ Parallel

▶ Concurrent

▶ Random

---

**Question No: 1 ( Marks: 1 ) - Please choose one**

In selection algorithm, because we eliminate a constant fraction of the array with each phase, we get the

▶ **Convergent geometric series (Page 37)**

▶ Divergent geometric series

▶ None of these

**Question No: 1 ( Marks: 1 ) - Please choose one**

Due to left-complete nature of binary tree, heaps can be stored in

▶ Link list

▶ Structure

▶ **Array (Page 40)**

▶ None of above

## **CS609- System Programming**

### **Midterm Quizzes (Quiz No.1 & 2)**

**Quiz No.1 (04 – MAY - 2013)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

The time assumed for each basic operation to execute on RAM model of computation is-----

Infinite

Continuous

**Constant (Page 10)**

Variable

**Question No: 1 ( Marks: 1 ) - Please choose one**

If the indices passed to merge sort algorithm are not equal, the algorithm may return immediately.

True

**False (Page 28)**

**Question No: 1 ( Marks: 1 ) - Please choose one** Brute-force algorithm uses no intelligence in pruning out decisions.

**True (Page 18)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

In analysis, the Upper Bound means the function grows asymptotically no faster than its largest term.

**True (Page 24)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

For small values of n, any algorithm is fast enough. Running time does become an issue when n gets large.

**True (Page 14)**

Fast

**Question No: 1 ( Marks: 1 ) - Please choose one**

The array to be sorted is not passed as argument to the merge sort algorithm.

True

**False**

**Question No: 1 ( Marks: 1 ) - Please choose one**

In simple brute-force algorithm, we give no thought to efficiency.

**True**  
False (Page 11)

**Question No: 1 ( Marks: 1 ) - Please choose one**

The ancient Roman politicians understood an important principle of good algorithm design that is plan-sweep algorithm.

True

**False (Page 27) [Divide and Conquer]**

**Question No: 1 ( Marks: 1 ) - Please choose one**

In 2d-space a point is said to be \_\_\_\_\_if it is not dominated by any other point in that space.

Member

Minimal

**Maximal (Page 11)**

Joint

**Question No: 1 ( Marks: 1 ) - Please choose one**

An algorithm is a mathematical entity that is dependent on a specific programming language.

True **False**

**Question No: 1 ( Marks: 1 ) - Please choose one**

The running time of an algorithm would not depend upon the optimization by the compiler but that of an implementation of the algorithm would depend on it.

**True (Page 13)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

$f(n)$  and  $g(n)$  are asymptotically equivalent. This means that they have essentially the same \_\_\_\_\_ for large  $n$ .

Results

Variables

Size

**Growth rates (Page 23)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

$8n^2 + 2n - 3$  will eventually exceed  $c \cdot n$  no matter how large we make  $c$ .

**True (Page 25)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

If we associate  $(x, y)$  integers pair to cars where  $x$  is the speed of the car and  $y$  is the negation of the price. High  $y$  value for a car means a \_\_\_\_\_ car.

Fast

Slow

Expensive

**Cheap** (Page 11)

**Question No: 1 ( Marks: 1 ) - Please choose one**

The function  $f(n) = n(\log n + 1)/2$  is asymptotically equivalent to  $n \log n$ . Here Upper Bound means the function  $f(n)$  grows asymptotically \_\_\_\_\_ faster than  $n \log n$ .

More

Quiet

**Not (Page 24)**

At least

**Question No: 1 ( Marks: 1 ) - Please choose one**

After sorting in merge sort algorithm, merging process is invoked.

Select correct option:

**True (Page 28)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

Asymptotic growth rate of the function is taken over \_\_\_\_\_ case running time.

Select correct option:

Best

Average

**Worst (Page 14)**

Normal

**Question No: 1 ( Marks: 1 ) - Please choose one**

In analysis of  $f(n) = n(n/5) + n - 10 \log n$ ,  $f(n)$  is asymptotically equivalent to \_\_\_\_\_.

n

2n

n+1

**n<sup>2</sup> (Page 23)**

**Question No: 1 (Marks: 1) - Please choose one**

Algorithm is concerned with.....issues.

Macro

Micro

**Both Macro & Micro (Page 8)**

Normal

**Question No: 1 (Marks: 1) - Please choose one**

We cannot make any significant improvement in the running time which is better than that of brute-force algorithm.

True

**False (Page 18)**

**Question No: 1 (Marks: 1) - Please choose one**

In addition to passing in the array itself to Merge Sort algorithm, we will pass in \_\_\_\_\_other arguments which are indices.

**Two (Page 28)**

Three

Four

Five

**Question No: 1 (Marks: 1) - Please choose one**

Consider the following Algorithm: Fun(n){ if (n=1) return 1 else return (n \* Fun(n-1)) } Recurrence for the above algorithm is:

$nT(n-1)+1$

$2T(n-1)+1$

$T(n-1)+cn$

**$T(n-1)+1$**

**Question No: 1 ( Marks: 1 ) - Please choose one**

In analysis, the Lower Bound means the function grows asymptotically at least as fast as its largest term.

**True (Page 24)**

False

**Question No: 1 ( Marks: 1 ) - Please choose one**

Efficient algorithm requires less computational.....

Memory

Running Time

**Memory and Running Time (Page 9)**

Energy

**Question No: 1 ( Marks: 1 ) - Please choose one**

The O-notation is used to state only the asymptotic \_\_\_\_\_ bounds.

Two

Lower

**Upper (Page 25)**

Both lower & upper

**Question No: 1 ( Marks: 1 ) - Please choose one**

For the worst-case running time analysis, the nested loop structure containing one “for” and one “while” loop, might be expressed as a pair of \_\_\_\_\_ nested summations.

1

**2 (Page 16)**

3

4

**Question No: 1 ( Marks: 1 ) - Please choose one**

Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their \_\_\_\_\_ coordinates.

**X (Page 18)**

Y

Z

X & Y

**Question No: 1 ( Marks: 1 ) - Please choose one**

Brute-force algorithm for 2D-Maxima is operated by comparing \_\_\_\_\_ pairs of points.

Two

Some

Most

**All (Page 18)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

The function  $f(n) = n(\log n + 1)/2$  is asymptotically equivalent to  $n \log n$ . Here Lower Bound means function  $f(n)$  grows asymptotically at \_\_\_\_\_ as fast as  $n \log n$ .

Normal

**Least (Page 23)**

Most

All

**Question No: 1 ( Marks: 1 ) - Please choose one**

The definition of Theta-notation relies on proving \_\_\_\_\_ asymptotic bound.

One

Lower

Upper

**Both lower & upper (Page 25) rep**

**Question No: 1 ( Marks: 1 ) - Please choose one**

In plane sweep approach, a vertical line is swept across the 2d-plane and \_\_\_\_\_ structure is used for holding the maximal points lying to the left of the sweep line.

Array

Queue

**Stack (Page 18)**

Tree

**Question No: 1 ( Marks: 1 ) - Please choose one**

Algorithm analysts know for sure about efficient solutions for NP-complete problems.

Select correct option:

True

**False (Page 9)**

**Quiz No.1 (2012)**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** The number of nodes in a complete binary tree of height h is

**$2^{(h+1)} - 1$  (Page 40)**

$2 * (h+1) - 1$

$2 * (h+1)$

$((h+1) ^ 2) - 1$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The analysis of Selection algorithm shows the total running time is indeed \_\_\_\_\_in n,

arithmetic

geometric

**linear (Page 37)**

orthogonal

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

A (an) \_\_\_\_\_ is a left-complete binary tree that conforms to the heap order

**heap (Page 40)**

binary tree

binary search tree

array

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** Analysis of Selection algorithm ends up with,

**T(n) (Page 37)**

$T(1 / 1 + n)$

$T(n / 2)$

$T((n / 2) + n)$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** For the sieve technique we solve the problem,

**recursively (Page 34)**

mathematically

precisely

accurately

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

A heap is a left-complete binary tree that conforms to the \_\_\_\_\_

increasing order only

decreasing order only

**heap order (Page 40)**

(log n) order

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** In which order we can sort?

increasing order only

decreasing order only

**increasing order or decreasing order (Page 39)**

both at the same time

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** Divide-and-conquer as breaking the problem into a small number of

pivot

Sieve

**smaller sub problems (Page 34)**

Selection

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** For the heap sort we store the tree nodes in

**level-order traversal (Page 40)**

in-order traversal

pre-order traversal

post-order traversal

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The sieve technique works in \_\_\_\_\_ as follows

**Phases (Page 34)**

numbers

integers

routines

## **CS502 - Fundamentals of Algorithms**

### **Quiz No.1 12-11-2012**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

We do sorting to,

keep elements in random positions

keep the algorithm run in linear order

keep the algorithm run in  $(\log n)$  order

**keep elements in increasing or decreasing order**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Heaps can be stored in arrays without using any pointers; this is due to the \_\_\_\_\_ nature of the binary tree,

**left-complete(Page 40)**

right-complete

tree nodes

tree leaves

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Sieve Technique can be applied to selection problem?

**True (Page 35)**

False

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In Sieve Technique we do not know which item is of interest

**True (Page 34)**

False

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In the analysis of Selection algorithm, we eliminate a constant fraction of the array with each phase; we get the convergent \_\_\_\_\_ series in the analysis,

linear

arithmetic

**geometric (Page 37)**

exponent

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

For the heap sort, access to nodes involves simple \_\_\_\_\_ operations.

**arithmetic (Page 41)**

binary

algebraic

logarithmic

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** Slow sorting algorithms run in,

**$T(n^2)$  (Page 39)**

$T(n)$

$T(\log n)$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In the analysis of Selection algorithm, we make a number of passes, in fact it could be as many as,

$T(n)$

$T(n/2)$

**$\log n$  (Page 37)**

$n/2 + n/4$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The sieve technique is a special case, where the number of sub problems is just

5

many

**1 (Page 34)**

few

**Question No: 1 of 10 (Marks: 1) - Please choose one**

How many elements do we eliminate in each time for the Analysis of Selection algorithm?

$(n / 2) + n$  elements

**$(n / 2)$  elements (Page 37)**

$n / 4$  elements

$2n$  elements

**Question No: 1 of 10 (Marks: 1) - Please choose one**

One of the clever aspects of heaps is that they can be stored in arrays without using any \_\_\_\_\_.

**pointers (Page 40)**

constants

variables

functions

**Question No: 1 of 10 (Marks: 1) - Please choose one**

How much time merge sort takes for an array of numbers?

$T(n^2)$

$T(n)$

$T(\log n)$

**$T(n \log n)$  (Page 40)**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The reason for introducing Sieve Technique algorithm is that it illustrates a very important special case of,

**divide-and-conquer (Page 34)**

decrease and conquer

greedy nature

2-dimension Maxima

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** In Sieve Technique we do not know which item is of interest

**True (Page 34) rep**

False

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Theta asymptotic notation for  $T(n)$  :

Set of functions described by:  $c_1g(n)$  Set of functions described by  $c_1g(n) \geq f(n)$  for  $c_1$  s

**Theta for  $T(n)$  is actually upper and worst case comp (Not sure)**

Set of functions described by:  $c_1g(n)$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Memoization is?

To store previous results for future use

**To avoid this unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later (page 74)**

To make the process accurate

None of the above

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Which sorting algorithm is faster

**O (n log n) Page 26**

O  $n^2$

O  $(n+k)$

O  $n^3$

O  $n^2$

O  $n^3$

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** Quick sort is

Stable & in place

**Not stable but in place (Page 54)**

Stable but not in place

Some time stable & some times in place

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** One example of in place but not stable algorithm is

Merger Sort

**Quick Sort (Page 54)**

Continuation Sort

Bubble Sort

**Question No: 1 of 10 (Marks: 1) - Please choose one**

Cont sort is suitable to sort the elements in range 1 to k

K is Large

K is not known

K may be small or large

**K is small (Page 57)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

In place stable sorting algorithm.

**If duplicate elements remain in the same relative position after sorting (Page 54)**

One array is used

More than one arrays are required

Duplicating elements not handled

**Question No: 1 of 10 (Marks: 1) - Please choose one**

Which may be a stable sort?

Merger

**Insertion (Page 54)**

Both above

None of the above

**Question No: 1 of 10 (Marks: 1) - Please choose one**

An in place sorting algorithm is one that uses \_\_\_ arrays for storage

Two dimensional arrays

More than one array

**No Additional Array (Page 54)**

None of the above

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Sieve Technique applies to problems where we are interested in finding a single item from a larger set of

\_\_\_\_\_

**n items (Page 34)**

phases

pointers

constant

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Sorting is one of the few problems where provable \_\_\_\_\_ bounds exists on how fast we can sort,

upper

**lower (Page 39)**

average

log n

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Counting sort has time complexity:

**O(n) (Page 58)**

O(n+k)

O(k)

O(nlogn)

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The running time of quick sort depends heavily on the selection of

No of inputs

Arrangement of elements in array

Size o elements

**Pivot elements (Page 49)**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Which may be stable sort:

Bubble sort

Insertion sort

**Both of above (Page 54)**

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

One Example of in place but not stable sort is

**Quick (Page 54)**

Heap

Merge

Bubble

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In Quick Sort Constants hidden in  $T(n \log n)$  are

Large

Medium

**Small** [Click here for detail](#)

Not Known

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Quick sort is based on divide and conquer paradigm; we divide the problem on base of pivot element and:

There is explicit combine process as well to conquer the solution.

No work is needed to combine the sub-arrays, the array is already sorted

Merging the sub arrays

**None of above. (Page 51)**

Ref: - random choices for the pivot element and each choice have an equal probability of  $1/n$  of occurring.  
So we can modify the above recurrence to compute an average rather than a max

---

## CS501 - Quiz No.2 (Spring 2013)

**Question No: 1 of 10 (Marks: 1) - Please choose one**

A point p in 2-dimensional space is usually given by its integer coordinate(s)\_\_\_\_\_

p.x only

p.y only

p.x & p.z

**p.x & p.y (Page 10)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

In \_\_\_\_\_ we have to find rank of an element from given input.

Merge sort algorithm

**Selection problem (Page 34)**

Brute force technique

Plane Sweep algorithm

**Question No: 1 of 10 (Marks: 1) - Please choose one**

In Heap Sort algorithm, if heap property is violated \_\_\_\_\_

We call Build heap procedure

**We call Heapify procedure**

We ignore

Heap property can never be violated

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Upper bound requires that there exist positive constants  $c_2$  and  $n_0$  such that  $f(n) \leq c_2 n$  for all  $n \leq n_0$  (ye

question ghalat lag raha hai mujhae

Less than

**Equal to or Less** (Page 25)  
**than** Equal or Greater  
than Greater than

**Question No: 1 of 10 (Marks: 1) - Please choose one**

A RAM is an idealized algorithm with takes an infinitely large random-access memory.

True

**False (Page 10)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

\_\_\_\_\_ is one of the few problems, where provable lower bounds exist on how fast we can sort.

Searching

**Sorting (Page )**

Both Searching & Sorting

Graphing

**Question No: 1 of 10 (Marks: 1) - Please choose one**

Floor and ceiling are \_\_\_\_\_ to calculate while analyzing algorithms.

Very easy

**Usually considered difficult (Page 31)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

In Heap Sort algorithm, the maximum levels an element can move upward is \_\_\_\_\_

**Theta (log n) (Page 43)**

Order (log n)

Omega (log n)

O (1) i.e. Constant time

**Question No: 1 of 10 (Marks: 1) - Please choose one**

A point p in 2-dimensional space is usually given by its integer coordinate(s)\_\_\_\_\_

p.x only p.y

only p.x & p.z

**p.x & p.y (Page 17)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

In Heap Sort algorithm, the total running time for Heapify procedure is \_\_\_\_\_

**Theta (log n) (Page 43)**

Order (log n)

Omega (log n)

O (1) i.e. Constant time

**Question No: 1 of 10 (Marks: 1) - Please choose one**

Algorithm is a mathematical entity, which is independent of a specific machine and operating system.

True

**False (Page 7)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

While Sorting, the ordered domain means for any two input elements x and y \_\_\_\_\_ satisfies only.

$x < y$

$x > y$

$x = y$

**All of the above** (Page 39)

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** Quick sort is best from the perspective of Locality of reference.

**True**  
False (Page 9)

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

Sorting can be in \_\_\_\_\_

Increasing order only

Decreasing order only

**Both Increasing and Decreasing order** (Page 39)

Random order

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In Heap Sort algorithm, we build \_\_\_\_\_ for ascending sort.

**Max heap**  
Min heap (Page 41)

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one** In Sieve Technique, we know the item of interest.

True

**False (Page 34)**

**Question No: 1 of 10 (Marks: 1) - Please choose one**

While solving Selection problem, in Sieve technique we partition input data \_\_\_\_\_

In increasing order

In decreasing order

**According to Pivot (Page 3**

Randomly

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

In pseudo code, the level of details depends on intended audience of the algorithm.

**True (Page 12)**

False

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

The sieve technique works where we have to find \_\_\_\_\_ item(s) from a large input.

**Single (Page 34)**

Two

Three

Similar

**Question No: 1 of 10 ( Marks: 1 ) - Please choose one**

If the indices passed to merge sort algorithm are \_\_\_\_\_, then this means that there is only one element to sort.

Small

Large

**Equal (Page 28)**

Not Equal

1) The sieve technique works where we have to find ----- item(s) from a large input.

• **Single** **page 34**

- Two
- Three
- Similar

2) The only way to convert an empty string into a string of  $j$  characters is by doing  $j$  insertions, represented as

- $E(i, j) = I$
- $E(i, 0) = i$
- **$E(0, j) = j$**  **page 78**
- $E(1, j) = j$

3) In Selection problem, the Sieve technique works in -----

- Non-recursive manner
- Constant time
- **Phases** **page 34**
- One complete go

4) Algorithm is a sequence of computational steps that ----- the input into output.

- Merge
- Assign
- **Transform** **page 7**
- Integrate

5) In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with ----- in the leaf tree.

- Second last
- **Last** **page 41**
- First
- Any

6) If  $p_j$  dominates  $p_i$  and  $p_i$  dominates  $p_h$  then  $p_j$  also dominates  $p_h$ . It means dominance relation is

- **Transitive page 18**
- Non transitive
- Equation

Symbolic

7) In the following code the statement “cout<<j;” executes ----- times

```
For(j=1;j<=5;j=j+2)
```

```
Cout<<j;
```

- 5 times
- 2 times
- **3 times (conceptual)**
- 0 times

9) For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from -----.

- Half of the input values
- **All possible random values page 50**
- Pivot is input separately
- Values greater than 5

10) In the statement, “output  $P[i].x, P[i].y$ ”, the number of times elements of P are accessed is -----.

- 1
- **2 page 14**
- 3
- 4

11) Identify the TRUE statement.

- The Knapsack problem does not belong to the domain of optimization problems.
- **The Knapsack problem belongs to the domain of optimization problems Page 91**
- The Knapsack problem cannot be solved by using dynamic programming.
- The Knapsack problem is optimally solved by using brute force algorithm.

12) Matrix multiplication is a(n) ----- operation.

- Commutative
- **Associative page 85**
- Neither commutative nor associative
- Commutative but not associative

13) ----- provides us more accurate result when input values are not closer to each other.

- Average

• **Median page 34(conceptual)**

14) In merge sort algorithm, we split the array around the ----- index q.

- Entering
- **Mid page 28**

- Exitin

Summing

14) In the analysis of Selection algorithm, we get the convergent ----- series.

- Harmonic
- Linear
- Arithmetic
- **Geometric page 37**

16) In Selection problem, the Sieve technique -----.

- Adds some more input items each time.
- Do not work recursively
- Do not uses Divide and Conquer approach
- **Eliminate undesired data items each time page 35**

17) If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then each entry in resultant matrix takes ----- time.

- **$O(q)$  page 84**

- $O(1)$
- $O(p \times q)$
- $O(q \times r)$

18) In Sorting the key value or attribute ----- form an ordered domain.

- Must be

**Not always** **page 39**

19) In Heap Sort algorithm, the total running time for Heapify procedure is ----- .

**Theta  $(\log n)$**  **page 43**

- Order  $(\log n)$
- Omega  $(\log n)$
- $O(1)$  i.e. Constant time

20) Pseudo code of algorithm are to be read by -----.

- **People** **page 12**
- RAM
- Computer
- Compiler

21) In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object 'i', then we gain -----.

- W (Total Weight of Knapsack)
- V (Total Value of all items)
- **$v_i$ (Value of object i)** **page 93**
- None of the given option

22) The worst case running time of Quick sort algorithm -----.

- Cannot be quadratic

**Is quadratic (google)**

- Is always exponential
- Is linear

23) The only way to convert a string of  $i$  characters into the empty string is with  $i$  deletions , represented as

- $E(0, j)=j$
- $E(i, j)=i$
- $E(0, i)=j$
- **$E(i, 0)=i$  page 78**

24) In Selection problem, the rank of an element will be its ----- position if we sort the input data.

- First
- **Final page 34**
- Second last
- Last

25) Which one is not passed as parameter in Quick sort algorithm?

- End of array
- Middle of array

**• Array (containing input elements) (cs301)**

- Start of the array

26) While Sorting, the ordered domain means for any two input elements  $x$  and  $y$  ----- satisfies only.

- $X < y$
- $X > y$

- X=y
- **All of the above** page 39

27) Which of the following is calculated with Big O notation?

- Medium bounds
- **Upper bounds** page 25
- Lower bounds
- Both upper and lower bounds

28) For solving Selection problem, we introduced Sieve technique due to -----

- Using Decrease and Conquer strategy
- Avoiding to sort all input data
- **Eliminating Rank of an element** page 35

Using Brute-Force approach

29) The worst-case running time of merge sort is ----- in order to sort an array of n elements.

- $O(\log n)$
- $O(n)$

**$O(n \log n)$  page 30**

•

30) Quick sort is a recursive algorithm.

- False
- **True (page49)**

31) While analyzing Selection algorithm, we make a number of passes , in fact it could be as many as -----.

- $n(n+1)$
- **$\log(n)$  page 37**

- $n/3$
- $n/4$

32) Approach for solving geometric problems by sweeping a line across the plane is called ----- sweep.

- Line
- **Plane page 18**
- Cube
- Box

33) ----- time is the maximum running time over all legal inputs.

- **Worst-case page 13**
- Average-case
- Best-case
- Good-case

34) The asymptotic growth of  $n(n+1)/2$  is:

- **$O(n^2)$  page 15**
- $O(n)$
- $O(n+2)$
- $O(n \log n)$

35) One of the limitation in 0/1 Knapsack is that an item can either be ----- in the bag or not.

- Use
- **Put page 91**
- Move
- Store

36) ----- is one of the few problems, where provable lower bounds exist on how fast we can sort.

- Searching
- **Sorting page 39**
- Both searching and sorting
- Graphing

37) Selection algorithm takes theta -----.

- $(n^2)$
- **(n) page 37**
- $\log(n)$
- $n\log(n)$

38) The Sieve technique is a special case, where the numbers of sub-problems is just -----.

- **1 page 34**
- 2
- 3
- 4

39) Following is not the application of Edit Distance problem

- Speech recognition
- Spelling Correction
- **Ascending Sort page 76**
- Computational Molecular Biology

40) We can use the optimal substructure property to devise a ----- formulation of the edit distance problem.

- Selective
- Optimum
- Iterative

**Recursive (p78)**

41) In plane sweep approach, a vertical line is swept across the 2d-plane form -----

- Right to left
- **Left to right** **page 18**
- Top to bottom
- Bottom to top

42) In 3-Dimensional space, a point P has ----- coordinate(s).

- (X,Y)
- (X,0)
- (0,Y)
- **(X,Y,Z)** **(google)**

43) ----- items are not allowed in the 0/1 Knapsack.

- Lighter
- **Fractional** **page 91**
- Whole
- Weighty

44) Radix sort is not a non-comparative integer sorting algorithm.

- True
- **False** **(google)**

45) Fibonacci sequence was named on -----, a famous mathematician in 12<sup>th</sup> Century.

- Fred Brooks
- Grady Booch
- **Leonardo Pisano** **page 73**
- Edgar F.Codd

46) In generating Fibonacci Sequence, we can avoid unnecessary repetitions by ----- process.

- Tokenization
- **Memoization page 74**
- Randomization
- Memorization

47) The sequence of merge sort algorithm is:

- Divide-Combine-Conquer
- Conquer-Divide-Combine
- **Divide-Conquer-Combine page 27**
- Combine-Divide-Conquer

48) Sorting can be in -----.

- Increasing order only
- Decreasing order only
- **Both Increasing and Decreasing order page 52**
- Random order

49) The process of ----- ends when you are left with such tiny pieces remaining that it is trivial to solve them.

- Brute-force
- Plan-sweep
- **Divide and Conquer page 27**
- Axis-sweep

50) Time complexity of the "0-1" knapsack algorithm depends on -----.

- Number of items
- Capacity of the knapsack
- Size of the table

• **Number of items and capacity of knapsack (Google)**

51) In max-heap, largest element is stored at root node. Where is the smallest element stored?

- Right Node

• **Leaf Node (Google)**

- Middle Node
- Left Node

52) For ----- values of n, any-----algorithm is fast enough.

• **Small pg#14**

- Medium
- Large
- Infinity

53) To predict the cost of an algorithm in terms of resources is called -----.

- Key

• **Analysis (Google)**

- Design
- Output

54) Dynamic programming comprises of -----

• **Recursion only pg#75**

- Repetition only
- Recursion and Repetition
- No Repetition but Recursion

55) For Quick sort algorithm, Partitioning takes theta -----?

- $(n)$
- $\log(n)$
- **$n\log(n)$  (Google)**
- $n^2\log(n)$

56) An in-place sorting algorithm is one that ----- uses additional array for storage.

- Always
- Permanently
- **Does not pg#54**
- Sometime

57) In Dynamic Programming approach, we do not store the solution to each sub-problem in case if it reappears.

- True
- **False pg#75**

58) Counting sort has time complexity:

- **$O(n+k)$  pg#58**
- $O(n)$
- $O(k)$
- $O(n\log n)$

59) Due to left complete nature of binary tree, the heap can be stored in

- **Arrays pg#40**
- Structures
- Link list
- Stack

60) For the heap sort we store the tree node in

- **Level-order traversal pg#40**

- In-order traversal
- Pre-order traversal
- Post-order traversal

61) In the clique cover problem, for two vertices to be in the same group, they must be -----each other.

- Apart from
- Far from
- Near to

• **Adjacent to pg#176**

62) In in-place sorting algorithm is one that uses arrays for storage :

**No additional array pg#54**

- An additional array
- Both of above may be true according to algorithm
- More than 3 arrays of one dimension

63) Brute-force algorithm for 2D-Maxima is operated by comparing ----- pairs of points.

- Two
- Some
- Most

**All pg#18**

64)  $F(n)$  and  $g(n)$  are asymptotically equivalent. This means they have essentially the same ----- for large  $n$ .

- Results

**Growth rates pg#23**

- Variables
- Size

65) Quick sort is

- Stable & in place

• **Not stable but in place pg#54**

- Stable but not in place
- Some time stable & some times in place

66) Heaps can be sorted in arrays without using any pointers; this is due to the ----- nature of the binary tree.

- **Left-complete pg#40**

- Right-complete
- Tree nodes
- Tree leaves

67) Which may be a stable sort?

- **Merger**

- **Insertion**

- **Both above page 54**
- None of the above

68) For the Sieve Technique we take time

- **$T(nk)$  page 34**
- $IT(n/3)$
- $n^2$
- $n/3$

69) One of clever aspects of heaps is that they can be stored in arrays without using any -----.

- **Pointers page 40**
- Constants
- Variables
- Functions

70) Counting sort is suitable to sort the elements in range 1 to k.

- K is large
- K is not known
- K may be small or large
- **K is small page 57**

71) Asymptotic growth rate of the function is taken over ----- case running time.

- Best
- **Worst** page 14
- Average
- Normal

72) The Sieve technique is a special case, where the number of a sub-problems is just .

- 5
- **1** page 34
- Few

73) Dynamic programming algorithms need to store the results of intermediate sub-problems.

- **True** page 75
- False

74) In counting sort, once we know the ranks, we simply ----- numbers to their final positions in an output array.

- Delete
- **Copy** page 57
- Mark
- Arrange

75) In Quick sort, we don't have control over the sizes of recursive calls.

- **True** page 49
- False
- Less information to decide
- Either true or false

76) Before sweeping a vertical line in a plane sweep approach, in start sorting of the points is done in increasing order of their ----- coordinates.

• **X (page #18)**

- Y
- Z
- X&Y

77) Random access machine or RAM is a/an

- Machine build by Al-Khwarizmi
- Mechanical machine

• **Mathematical model (page #10)**

- Electronics machine

78) The Huffman codes provide a method of encoding data inefficiently when coded using ASCII standard .

- True

• **False(page #99)**

79) A heap is a left-complete binary tree that conforms to the -----.

- Increasing order only
- Decreasing order only

• **Heap order (page# 40)**

- $(\log n)$  order

80) For the sieve technique we solve the problem,

• **Recursively (page #34)**

- Mathematically
- Precisely
- Accurately

81) Median is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

- True
- **False (page # 34)**

82) The omega-notation allows us to state only the asymptotic-----bounds.

- Middle
- **Lower (page #25)**
- Upper
- Both middle and upper

83) In plane sweep approach, a vertical line is swept across the 2-D plane and-----structure is used for holding the maximal points lying to the left of the sweep line.

- Array
- Queue
- **Stack (page #18)**

- Tree

84) When a heapify procedure is applied to the root node to restore the heap, then at each level, the comparison performed takes time.

- It will take  $O(\log n)$
- It cannot be predicted
- **It will take  $O(1)$  (page #43)**
- Time will vary according to the nature of input data

85) Efficient algorithm requires less computational-----.

- Memory
- Running time
- **Memory and running time (page #9)**
- Energy

86) Recurrence can be described in terms of a tree.

- **Yes (page#31)**
- No

87) Time complexity of dynamic programming based algorithm for computing the minimum cost of Chain Matrix multiplication is-----.

- $\log n$
- $n$
- $n^2$  (n square)

•  **$n^3$  (n cube) (page#90)**

88) In order to say anything meaningful about our algorithms, it will be important for us to settle on a-----  
-----.

- Java programming
- C++ programming
- Pseudo program

• **Mathematical model of computation (page # 10)**

89) The iteration method is used for -----.

- Comparing sorting algorithms only

• **Solving recurrence relation (page # 31)**

- Merging elements in merging sort
- Divide elements in merge sort

90) Chain matrix multiplication problem can be solved through-----strategy.

• **Dynamic programming (page #85)**

- Greedy
- Divide and conquer
- Sorting

91) For comparison-based sorting algorithms, it is -----possible to sort more efficiently than  $\Omega(n \log n)$  time.

- Always

• **Not (page #54)**

- Sometimes not
- Sometimes

92) Dynamic Programming approaches are usually useful in solving optimization problems.

• True (page # 97)

- False

93) In sorting the key value or attribute-----from an ordered domain.

• Must be (page #39)

- Not always

94) Result of asymptotical analysis of  $n(n-3)$  and  $4n*n$  is that-----.

- $n(n-1)$  is asymptotically less
- $n(n-1)$  is asymptotically Greater
- both are asymptotically Not equivalent

• both are asymptotically Equivalent (page #23)

95) Floor and ceiling are-----to calculate while analyzing algorithms.

- Very easy

• Usually consider difficult pg#31

96) ----- of reference is an important fact of current processor technology.

- Defining
- Assigning
- Formality

• Locality (page # 8)

97) As per algorithm of dynamic programming , we need to store

- First sub- problem only
- Best solution only

• Intermediate sub-problems (page #75)

- Final solution only

8) In Sieve technique, we solve the problem-----.

• In recursive manner (page # 34)

- Non recursively
- Using Merge sort algorithm
- Using Brute force technique

99) Upper bound requires that there exist positive constant  $c_2$  and  $n_0$  such that  $f(n) \leq c_2 n$  for all  $n \geq n_0$ .

- Less than
- **Equal to or less than (page # 24)**
- Equal to or greater than
- Greater than

100) In 2-D matrix problem, A point  $p$  is said to be dominated by point  $q$  if-----.

- $P.x = q.x$  and  $p.y = q.y$   
 $P.x \leq q.x$  and  $p.y \leq q.y$
- **$P.x \leq q.x$  and  $p.y \leq q.y$  (page # 11)**

101) A Random Access Machine (RAM) is an idealized machine with----- random access.

- **Infinite large (page # 10)**

- 512 MB
- 256 MB
- 2 GBs

102) There are-----entries in the Edit Distance Matrix.

- $\Theta(n)$
- **$\Theta(n^2)$  (page # 84)**
- $\Theta(n+2)$
- $\Theta(n+100)$

103) Counting sort is suitable for sorting the elements within range 1 to  $P$ , where-----.

- $P$  is large
- **$P$  is small (page # 57)**
- $P$  is very large
- $P$  is undetermined

104) While analyzing searching algorithm, we make a number of passes, in fact it called----

- $n(n+1)$
- **$\log(n)$  (page #37)**
- $n/3$

- $n/4$

105) It is not a Fibonacci Sequence.

1,1,1,2,3,5,8,13,21,34,

55,----

- **True (page #73)**

- False

106) In Random Access Machine (RAM), instructions are executed in-----.

- Parallel
- Batch

**One by one (page # 10)**

- Multiple times

107) Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in a(n):

- Array

**Table (page #86)**

- Variable
- Class

108) An algorithm is a mathematical entity, which is independent of-----.

- Programming language
- Machine and programming language
- Compiler and programming language

**Programming language Compiler and Machine (page #7)**

109) In Fibonacci sequence, unnecessary repetitions do not exist at all.

**True (page# 74)**

- False

110) In Quick sort algorithm, pivots form-----.

- Stack

**Queue**

- Graph

111) In Heap Sort Algorithm, Heapify procedure is----- in nature.

**Recursive (page #43)**

- Non-recursive

112) In Bucket sort, if there are duplicates then each bin can be replaced by a :

- Stack

**Link list (page #69)**

- Hash table
- Heap

113) Counting sort assumes that the numbers to be sorted are in the range -----.

- k to n where n is large

**1 to k where k is small (page # 57)**

- k to n where k is small
- k to n where n is small

114) In Dynamic Programming, our approach is to -----.

- Develop the solution in a top-down fashion.
- Express the problem non-recursively.

**○ Build the solution in a bottom-up fashion. Pg#92**

- Input several sub-problems simultaneously.

115) In knapsack Problem, the goal is to put items in the knapsack such that the value of the items is -----

----- subject to weight limit of knapsack.

- Minimized
- Decreased

**Maximized**

- None of the given option

116) If there are  $\Theta(n^2)$  entries in edit distance matrix then each entry  $E(i,j)$  takes ----- time to compute.

- $\Theta(n)$
- $\Theta(n^2)$

**$\Theta(1)$  (page #84)**

- $\Theta(n \log n)$

117) In ----- knapsack Problem, limitation is that an item can either be put in the bag or not.

Fractional items are not allowed.

- 0
- 1

**0/1 (page #91)**

- Fractional

118) Heapsort is a/an ----- and ----- sorting algorithms.

- Not in-place, not stable one
- In-place, stable one

**In-place, not stable one (page#54)**

- Not in-place, stable one

119) Selection sort is not an in-place sorting algorithm.

- True

**False (page #54)**

120) Memoization is a part of Dynamic Programming strategy.

**True (page 74)**

- False

127) Quick sort algorithms divide the entire array into-----sub arrays.

- One

**Two (Google)**

- Three
- Four

121) Insertion sort is an in-place algorithm.

**a) True (page #54)**

b) False

122) When matrix A of 5x3 is multiply with matrix B of 3x4 then the number of multiplication required is :

- 15
- 12
- 36
- **60 page 84**

123) If matrix A of dimensions 2x4 is multiplied with matrix B of dimensions 4x3, then the dimensions of the resultant matrix is :

- 2x4
- 4x3
- 3x4
- **2x3 page 84(conceptual)**

124) Dynamic Programming strategy is useful when sub-problems are independent.

- True
- **False google**

125) If Matrix-A of dimension 2x4 multiplied with Matrix-B of dimension 4x3, then the dimension of the resultant Matrix is:

- 2x4
- 4x3
- 3x4
- **2x3 page 84(conceptual)**

126) In the following code the statement "cout<< i" executes-----

```
---times. for (int i=1; i<=n; i++)
```

```
 cout<< i;
```

- **n times (conceptual)**
- n+5 times
- Infinite times
- Zero times

127) Edit distance algorithm based on-----strategy.

- Greedy
- **Dynamic programming page 77**
- Divide and conquer
- Searching

128) If Matrix A has dimensions "3x2" and Matrix B has dimensions "2x3", then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.

- 3x2
- 2x3
- 2x2
- **3x3 page 84(conceptual)**

129) The function  $f(n) = n(\log n + 1)/2$  is asymptotically equivalent to  $n \log n$ . Here Lower Bound means function f

(n) grows asymptotically at ----- as fast as  $n \log n$ .

- **Least page 24**
- Normal
- Most
- All

130) Single atom from a larger set of -----.

- Constant
- Pointers
- Phases
- **n items page 34**

131) Fibonacci sequence was named on a-----, a famous mathematician in 12<sup>th</sup> century.

- Fred brooks
- Grady booch

**Leonardo Pisano (page # 73)**

- Edgar F.Codd

132) In dynamic programming, our approach is to-----.

- Develop the solution in a top down fashion
- Express the problem non-recursively

**Build the solution in a bottom up fashion (page #92)**

- Input several sub-programs simultaneously

133) In heap sort algorithm, we build-----for ascending order.

- **Max heap page 42(conceptual)**
- Min heap

134) Bubble sort is not an in-place sorting algorithm.

- True

**False (page #54)**

135) In partition algorithm, the subarray-----has elements which are greater than pivot element X.

- $A[p\dots r]$
- $A[p\dots q-1]$
- $A[q]$

**$A[q+1\dots r]$  (page #46)**

136) In Heap sort algorithm, if heap property is violated-----.

- We call build heap procedure

**We call heapify procedure (page # 41)**

- We ignore
- Heap property can never be violated

137) -----is not a characteristics of Random Access Machine.

- Single-processor
- Assigning a value to a variable

**Locality of reference (page #10)**

- Executing an arithmetic instruction

138) Boolean operation is a-----operation on an idealized RAM model of computation.

- Starting

**Basic (page #10)**

- Advance
- Normal

139) How much time merge sort takes for an array of numbers?

- $T(n^2)$
- $T(n)$
- $T(\log n)$

•  **$T(n \log n)$  page 4**

140) Sorting is one of the-----problems in Computer Sciences.

**Easiest page 39**

- Solution-less
- Least popular
- Tony Hoare

142) We can make-----recursive calls in Fibonacci sequence.

- **Infinite** **google**
- Finite

140) Insertion sort is a ----- sorting algorithm.

- Unstable
- **In-place** **page 54**
- Not in-place
- In-portion

141) We do not need to prove comparison- based sorting algorithms by mathematically , it always takes-----

-----;

- Big Oh  $n \log(n)$
- **Omega  $n \log(n)$**  **page 54**
- Omega  $n(n^2)$
- Theta  $n \log(n)$

142) A Principal operation for maintaining the heap property is called heapify, it is also called :

- **Shifting Down** **page 43**
- Shifting Up
- Shifting Right
- Shifting Left

143) The number of nodes in a complete binary tree of height h is:

- $2^{(h+1)}-1$
- $2^{(h+1)}-1$
- **$2^{(h+1)}$**  **page 45**
- $((h+1)^2)-1$

144) After partitioning array in Quick sort, pivot is placed in a position such that:

- Pivot is the first element of array

- Pivot is the last element of array
- **Values smaller than pivot are on left and larger than pivot are on right**  
**page 35**

- Values larger than pivot are on left and smaller than pivot are on right

145) ----- programming is essentially recursion without repetition:

- Fast
- **Dynamic** **page 75**
- Array
- $N \log n$

146) We can use the \_\_\_\_\_ property to devise a recursive formulation of the edit distance problem.

- small substructure
- algorithmic

real

- **optimal substructure** **page 78**

147) Applying the sieve technique to selection problem, \_\_\_\_\_ element is picked from array.

- Output
- Total
- Input
- **Pivot** **page 35**

148) If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then the dimension of the resultant matrix is:

- $p \times q$
- **$p \times r$**  **page 84**
- $q \times r$
- $r \times p$

149) In knapsack problem, each item must be entirely accepted or rejected, is called \_\_\_\_\_ problem.

- Fractional
- **"0-1"** **page 92**
- Linear
- Optimal

150) In the case of \_\_\_\_\_ analysis does not depend upon on the distribution of input.

- Merge sort
- **Quick sort** **page 50**
- Insertion sort
- Heap sort

151) In Heapsort Algorithm, total time taken by heapify procedure is:

- **$O(\log n)$  page 43**
- $O(\log^2 n)$
- $O(n \log n)$
- $O(n^2 \log n)$

152) In merge sort algorithm, we split the array \_\_\_\_\_ to find index q.

- from start
- **midway page 28**
- from end
- both from start or end

153) Al-Khwarizmi was a/an \_\_\_\_\_

- artist
- **mathematician page 7**
- astronomer
- khalifah

154) In Sieve Technique, we know the item of interest.

- True
- **False page 34**

155) Radix sort is a \_\_\_\_\_ integer sorting algorithm.

- Comparative
- **Non-comparative google**
- In-Place
- Unstable

156) Random Access Machine (RAM) can execute \_\_\_\_\_ instruction.

- only logical
- parallel
- only arithmetic
- **logical and arithmetic page 10**

157) Which of the following is calculated with Big Omega notation?

- Medium bounds
- **Upper bounds page 25**
- Lower bounds
- Both upper and lower bounds

158) Asymptotic growth of  $8n^2 + 2n - 3$  is:

- $O(n^2+n)$
- **$O(n^2)$  page 23**
- $O(Bn^2)$
- $O(Bn^2+2n)$

159) \_\_\_\_\_ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.

- Loop
- **Memoization page 74**
- Recursion
- Function

160) In Heap Sort algorithm, the first step is to \_\_\_\_\_.

- **Call Build-Heap procedure page 43**
- Sort the array in descending order
- Call Heapify procedure
- Find the number of input elements

161) The time assumed for each basic operation to execute on RAM model of computation is \_\_\_\_\_.

- Infinite
- Continuous
- **Constant page 10**
- Variable

162) If input "n" is odd, then median will be \_\_\_\_\_.

- **$(n+1)/2$  page 34**

- $n/2$
- $(n-1)/2$
- $n+2$

163) In quick sort algorithm, \_\_\_\_\_ decides nature of Binary Search Tree formed by pivots.

- middle element from input
- smallest element from input
- **rank of the pivot page 49**
- largest element from input

164) The brute-force algorithm for 2D-Maxima runs in order O(-----) time:

- $n$
- $n(\log n)$

**$n^2$  page 18**

- $n^3$

165) In average-case time analysis of quick sort algorithm, the most balanced case for partition is when we divide

the list of elements into-----

- Equal no. of pieces as input elements
- Single piece exactly
- **Two nearly equal pieces google**
- Three nearly equal pieces

166) The main purpose of mathematical analysis is measuring the ----- required by the algorithm:

- Space
- **Execution time page 13**
- Inputs & outputs
- Execution time and memory

167) Dynamic Programming algorithms often use some kind of ----- to store the results of intermediate Sub-problems:

- **Table page 75**
- Variable
- Stack
- Loop

168) The average case running time of quick sort algorithm is theta -----:

- $\log(n)$
- $(n)$
- **$n \log(n)$  page 53**
- $n \times n \log(n)$

169) What is the best case time complexity of merge sort?

- $O((n^2 \log n))$
- $O((n \log^2))$
- **$O((n \log n))$  page 40**
- $O((n^2))$

170) While solving Selection problem, in Sieve technique we partition input data -----:

- In increasing order
- In decreasing order
- **According to Pivot page 35**
- Randomly

171) In average-case time the probability of seeing input is denoted by-----:

- $P\{I\}$
- $P[I]$

- **P(I) page 13**
- P<I>

172) To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is -----:

- Big-oh notation
- Theta notation
- Omega notation
- **All of the given google**

173) Asymptotic notations are used to describe ----- of an algorithm:

- Length
- **Running time (google)**
- Size
- Compile time

174) In recursive formulation of Knapsack Problem

$V[0,j] = \text{-----}$  for  $j \geq 0$

- -1
- **0 (page #93)**
- 1
- 2

175) If a matrix has three rows and two columns, then dimensions of matrix will be:

- **3x2 (google0)**
- 2x2
- 2x3
- 3x3

176) Selection sort takes theta --- in the worst case:

**(n<sup>2</sup>)**

- (n)
- Log(n)
- nlog(n)

177) Which of the following problems can be solved using dynamic problem?

- Bubble sort problem
- **Matrix chain multiplication problem page 92**
- Greedy search problem
- Fractional knapsack problem

178) ----- algorithm based on Dynamic Programming strategy:

- Heap sort
- Binary tree
- Quick sort
- **Edit distance page 77**

179) The definition of theta-notation relies on proving ----- asymptotic bound:

- One
- Lower
- Upper
- **Both lower and upper page 25**

180) In order to say anything meaningful about our algorithms, it will be important for us to settle on a -----:

- Java program
- C++ program
- Pseudo program
- **Mathematical model of computation page 10**

181) In asymptotical analysis of  $n(n-3)$  and  $4n*n$ , as  $n$  becomes large, the dominant (fastest growing) term is some Constant times-----,

- $n+1$
- $n-1$
- $n$
- **$n*n$  page 23**

182) ----- is in- place sorting algorithm

- **Bubble sort page 54**
- Merge sort
- Linear sort
- Binary sort

183) In chain matrix multiplication, if there are ----- ways in which outer most pair of parentheses can placed:

- $n^2$
- $2n$
- $n+1$
- **$n-1$  page 85**

184) In chain matrix multiplication, table is filled ----- to find the multiplication of matrix:

- row wise
- column wise
- **diagonally page 86**

185) Which one sorting algorithm is best suited to sort an array of 2 million elements?

- Bubble sort
- Insertion sort
- Merge sort
- **Quick sort google**

186) The problem with the brute-force algorithm is that it uses ----- in pruning out decisions:

- Worst-case time
- **No intelligence page 18**

- Outside looping
- Artificial intelligence

187) Quicksort is a/an ----- and ----- sorting algorithm:

- No in-place , not-stable one  
**In-place , not stable one page 54**

- In-place , stable one
- Not in-place , stable one

191) In Heap Sort algorithm, the maximum levels an element can move upward is -----:

- **Theta (log n) page 43**

- Big-oh(log n)
- Omega (log n)
- O (1) i.e. Constant time

192) If the time complexity of an algorithm is  $O(n)$  , then it is called ----- time complexity:

- **Linear page 37**
- Constant
- Average
- Exponential

193) Dynamic Programming approach is usually useful in solving ----- problems:

- Normal
- **Optimization page 97**
- Array
- Loop

194) What time does Merge Sort algorithm take in order to sort an array of 'n' numbers?

- $\Theta(n)$
- $\Theta(\log n)$
- $\Theta(n^2)$
- **$\Theta(n \log n)$  page 30**

## CS-502 (Fundamentals of Algorithms)

### Grand Quiz - Fall 2020

**If you found any error in the file, Please correct it accordingly.**

163) The sequence of merge sort algorithm is:

- Divide Combine-Conquer
- Conquer-Divide-Combine
- c. **Divide-Conquer-Combine Page 27**
- Combine-Divide-Conquer
- In \_\_\_\_\_ Knapsack Problem, limitation is that an item can either be put in the bag or not. Fractional items are not allowed.

0

1

- c. **0/1 Page 91**

Fractional

165) In Selection algorithm, we assume pivot selection takes theta \_\_\_\_\_ running time.

a. n

Page - 36

n<sup>2</sup>

n<sup>3</sup>

log (n)

- In Heap Sort algorithm (using max heap), when every time maximum elements removed from top \_\_\_\_\_.

We call merge Sort Algorithm

it becomes Order n<sup>2</sup> Algorithm

Divide and Conquer strategy helps us

d. We are left with a hole

Page – 41

- 167) If matrix A of dimension p x q is multiply with matrix B of dimension q x r, then each entry in resultant matrix takes \_\_\_\_\_ time.

a. O (q)

Page - 84

168) O (1)

169) O (p x q)

170) O (q x r)

169) \_\_\_\_\_ is a method of solving a problem in which we check all possible solutions to the problem to find the solution we need.

- Plane-Sweep Algorithm
- Sorting Algorithm

c. **Brute-Force Algorithm** **google**

- Greedy approach

171) The worst case running time of Quick sort algorithm \_\_\_\_\_.

- Cannot be quadratic

b. **Is quadratic** **google** [QuickSort and its Analysis \(codesdope.com\)](#) &

<http://personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Sorting/quickSort.htm>

Is always Exponential

Is linear

172) In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with \_\_\_\_\_ leaf in the tree.

- second last

b. **Last** **Page -41**

- First
- Any

174) Quick sort algorithm was developed by -

AlferdAho

Sedgewick

John Vincent Atanasoff

d. **Tony Hoare** **- Google wikipedia**

- If Matrix-A has dimensions "3x2" and Matrix-B has dimensions "2x3", then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.

3x2

2x3

2x2

**d. 3x3**

<http://www.calcul.com/show/calculator/matrix-multiplication>

- For comparison-based sorting algorithms, it is possible to sort more efficiently than Omega  $n \log(n)$  time.
  - a. Always
  - b. **Not** **P-54**
  - c. Sometimes

Sometimes not

176) Dynamic Programming approach is usually useful in solving optimization problems.

a. True

[https://en.wikipedia.org/wiki/Dynamic\\_programming](https://en.wikipedia.org/wiki/Dynamic_programming) &

<http://web.mit.edu/15.053/www/AMP-Chapter-11.pdf>

178) False

179) In Sorting the key value or attribute \_\_\_\_\_ from an ordered domain.

a. Must be page 39

- Not always
- May be
- Occasionally

181) Result of asymptotical analysis of  $n(n-3)$  and  $4n*n$  is that \_\_\_\_\_

- $n(n-1)$  is asymptotically Less
- $n(n-1)$  is asymptotically Greater
- Both are asymptotically Not equivalent

d. Both are asymptotically Equivalent page 23 ( $4n*n = 4n^2$ )

182) Floor and ceiling are \_\_\_\_\_ to calculate while analyzing algorithms a. Very easy

b. Usually considered difficult

P-31

3rd Option is missing

4th Option is missing

184) \_\_\_\_\_ of reference is an important fact of current processor technology.

- Defining
- Assigning

- Formality

**d. Locality**

**P-8**

185) In max-heap, largest element is stored at root node. Where is the smallest element stored?

- Right Node

**Leaf Node**

- Middle Node
- Left Node

- In average-case time analysis of Quick sort algorithm, the most balanced case for partition is when we divide the list of elements into \_.

**Equal no. of pieces as of input elements**

Single piece exactly

Two nearly equal pieces

Three nearly equal pieces

- Which of the following is calculated with Big O notation?

Medium bounds

b. **Upper bounds** **Page - 25**

- Lower bounds
- Both upper and lower bounds

187) Edit distance algorithm based on \_\_\_\_\_ strategy

- Greedy

b. **Dynamic Programming** **Page - 81**

## Divide and Conquer

### Searching

- In Heapsort Algorithm, total time taken by heapify procedure is \_\_\_\_\_

a.  $O(\log n)$  Page-43

- $O(\log^2 n)$
- $O(n \log n)$
- $O(n^2 \log n)$

193) Al-Khwarizmi was a/an \_\_\_\_\_

- Artist

b. **Mathematician**

**P-7**

Astronomer

Khalifah

194) When matrix A of  $5 \times 3$  is multiplied with matrix B of  $3 \times 4$  then the number of multiplications required is: Not found exactly

15

12

36

**60 Not Found exactly but as per formula at page 84,**

24. Pseudo code of algorithms are to be read by \_\_\_\_\_.

a. **People** **Page -12**

RAM

Computer

Compiler

195) The sieve technique is a special case, where the number of sub-problems is Just

\_\_\_\_\_

a. **1** **P-34**

b. 2

c. 3

d. 4

26. When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has \_\_\_\_\_ sub-problems.

a. **Overlapping** **– Google Search**

b. Over costing

c. Optimized

d. Three

27. Sieve technique is very important special case of Divide-and-Conquer strategy.

a. **True** **P-34**

b. False

28. In order to say anything meaningful about our algorithms, it will be important for us to settle on a \_\_\_\_\_.

a. Java Program

- b. C++ Program
- c. Pseudo program

**d. Mathematical model of computation**

**P-  
1  
0**

29. Merge sort is based on \_\_\_\_\_.

- a. Brute-force
- b. Plan-sweep
- c. Axis-sweep

**d. Divide and Conquer**

**P-  
27**

30. What time does Merge Sort algorithm take in order to sort an array of 'n' numbers?

- a. (n)
- b. (log n)
- c. (n<sup>2</sup>)

d. (n log n) Google Search

31. In Heap Sort algorithm, the first step is to \_\_\_\_\_ . a. Call Build-Heap

a. procedure Page - 46

- b. Sort the array in descending order
- c. Call Heapify procedure
- d. Find the number of input elements

32. The definition of theta-notation relies on proving \_\_\_\_\_ asymptotic bound.

- a. One
- b. Lower
- c. Upper

b. d. Both lower & upper Page - 25

33. In merge sort algorithm, to merge two lists of size n/2 to a list of size n, takes \_\_\_\_\_ time.

a. Theta (n) Page - 32

- b. Theta log(n)
- c. Theta log<sub>2</sub>(n)
- d. Theta n log(n)

34. We can make \_\_\_\_\_ recursive calls in Fibonacci Sequence.

- a. Infinite
- b. **Finite** google
- c. Only one
- d. Zero

35. Following is NOT the application of Edit Distance problem.

- a. Speech recognition
- b. Spelling Correction
- c. **Ascending Sort** Page - 76
- d. Computational Molecular Biology

36. In plane sweep approach, a vertical line is swept across the 2d-plane and structure is used for holding the maximal points lying to the left of the sweep line.

a. Array

b. Queue

c. **Stack** Page - 18

d. Tree

37. When a heapify procedure is applied to the root node to restore the heap, then at each level, the comparison performed takes time:

a. It will take  $(\log n)$ .

b. It can not be predicted

c. **It will take  $O(1)$ .** Page - 43

d. Time will vary according to the nature of input data.

38. \_\_\_\_\_ time is the maximum running time over all legal inputs.

a. **Worst-case** Page - 13

b. Average-case

c. Best-case

d. Good-case

39. Efficient algorithm requires less computational...

a. Memory

b. Running Time

c. **Memory and Running Time** Page - 9

d. Energy

40. For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from \_\_\_\_\_

a. half of the input values

b. all possible random values Page - 50

c. Pivot is input separately

d. values greater than 5

41. Selection algorithm takes theta \_\_\_\_\_

a.  $(n^2)$

b.  $(n)$  Page - 37

c.  $\log(n)$

d.  $n \log(n)$

42. Recurrence can be described in terms of a tree.

a. **Yes** **Page - 31**

b. No

43. Time complexity of Dynamic Programming based algorithm for computing the minimum cost of Chain Matrix Multiplication is \_\_\_\_\_

a. Log n

b. n

c.  $n^2$  (n square)

d.  **$n^3$  (n cube)** **Page -90**

44. The Iteration method is used for \_\_\_\_\_

a. Comparing sorting algorithms only

b. **Solving Recurrence relations** **Page 31**

c. Merging elements in Merge sort

d. Dividing elements in Merge sort

45. In 3-Dimensional space, a point P has \_\_\_\_\_ coordinate(s).

a. (X, Y)

b. (X, 0)

c. (0, Y)

d. **(X, Y, Z)**

46. Chain matrix multiplication problem can be solved through \_\_\_\_\_ strategy.

a. **Dynamic programming** **Page - 85**

- b. Greedy
- c. Divide and conquer
- d. Sorting

47. Merge sort have running time....running time of Heap sort. Not found exactly

- a. Greater than
- b. **Less than** **Google**
- c. Equal to
- d. Different than

48. Median is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

- a. True
- b. **False** **Page – 34**

49. We do not need to mathematically prove that for comparison-based sorting algorithms always takes Omega  $n \log(n)$  time.

- a. **True** **Google & VU Tech** **(pg 46 not very clear)**
- b. False

50. The Omega-notation allows us to state only the asymptotic \_\_\_\_\_ bounds.

- a. Middle
- b. **Lower** **Page 25**
- c. Upper
- d. Both lower & upper

51. Both lower & upper Sorting can be in \_\_\_\_\_

- a. Increasing order only

b. Decreasing order only

**Both Increasing and Decreasing order – (My opinion)**

c. Random order

52. Radix sort performs sorting the numbers \_\_\_\_\_ digit (s) at a time.

a. **One** **Page - 71**

b. Two

c. Three

d. All

53. Quicksort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm.

a. Not in place, not stable one

b. **In place , not stable one** **Page - 54**

c. In place , stable one

d. Not in place , stable one

54. Consider three matrices X,Y,Z of dimensions 1x2, 2x3,3x4 respectively. The number of multiplications of (XY) Z is:

a. **18** **As per lecture slides**

b. 32

c. 24

d. 30

55. In Fibonacci Sequence, unnecessary repetitions do not exist at all.

a. True

b. **False** **Page – 74**

56. It is not a Fibonacci sequence . 1,1,1,2,3,5,8,13,21,34,55,.....

a. **True** **Page - 73**

b. False

57. Heap sort is a/ an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm.

a. Not in place, not stable one

b. **In place , not stable one** **Page - 54**

c. In place , stable one

d. Not in place , stable one

58. Identify the True Statement

a. The knapsack problem does not belong to the domain of optimization problems.

b. **The knapsack problem belongs to the domain of optimization problems.** **Page - 91**

c. The Knapsack problem cannot be solved by using dynamic programming

d. The knapsack problem is optimally solved by using brute force algorithm.

59. In Dynamic Programming, our approach is to \_\_\_\_\_

a. Develop the solution in a top-down fashion

b. Express the problem non-recursively

c. **Build the solution in a bottom-up fashion** **Page - 75**

d. Input several sub-problems simultaneously

60. Counting sort is suitable to sort the elements in range 1 to K;

a. K is large

b. **K is small** Page - 57

c. K may be large or small

d. None

61. We can multiply two matrices A and B only when they are compatible which means

a. **Number of columns in A must be equal to number of rows in B.**

**it seems Correct as per page 84**

**([https://www.varsitytutors.com/hotmath/hotmath\\_help/topics/matrix-multiplication#:~:text=You%20can%20only%20multiply%20two,an%20m%C3%97p%20matrix.](https://www.varsitytutors.com/hotmath/hotmath_help/topics/matrix-multiplication#:~:text=You%20can%20only%20multiply%20two,an%20m%C3%97p%20matrix.))**

b. Number of rows and columns do not matter

c. Number of columns in A must be equal to number of columns in B

d. Number of rows in A must be equal to number of rows in B

62. Matrix multiplication is a (n) \_\_\_\_\_ operation.

a. Commutative

b.

c. **b. Associative Page 85**

c. Neither commutative nor associative

d. Commutative but not associative

63. In Dynamic Programming approach, solution is modified / changed

a. Always once

b. **At each stage** google and wikipedia

c. Only for specific problems

- d. At 4<sup>th</sup> stage only
64. In Knapsack problem, the goal is to put items in the Knapsack such that the value of the items is \_\_\_\_\_ subject to weight limit of knapsack.
- a. Minimized
  - b. Decreased
  - c. **Maximized** Page - 91
  - d. None of the given options
65. An in-place sorting algorithm is one that \_\_\_\_\_ uses additional array for storage.
- a. Always
  - b. Permanently
  - c. **Does not** Page - 54
  - d. Sometime
66. Memoization is a part of Dynamic Programming Strategy.
- a. **True** Page - 74
  - b. False
67. If matrix A of dimension 2x4 is multiply with matrix B of dimension 4x3, then the dimension of resultant matrix is      Not found exactly
- a. 2x4
  - b. 4x3
  - c. 3x4
  - d. **2x3** It seems correct as per second last Para of page 84

68. In Dynamic Programming approach, we do not store the solution to each sub-problem in case if it reappears.

a. True

b. **False** Page - 75

69. Dynamic Programming is a problem-solving approach in which\_\_\_\_

a. Problem is solved in Zero time

b. Solution is developed only at final stage

c. Both are correct

d. **Both are incorrect** google

70. In Fibonacci sequence, each term is calculated by\_\_\_\_ previous\_\_ terms.

a. Subtracting, Two

b. Adding, Three

c. **Adding, Two** Page - 73

d. Multiplying, Two

71. Selection sort is not an in-place sorting algorithm.

a. **True** Page - 54

b. False

72. If there are  $\theta(n^2)$  entries in edit distance matrix then the total running time is:

a.  $\theta(n)$

b.  $\theta(1)$

c.  **$\theta(n^2)$**  Page - 84

d.  $\theta(n \log n)$

73. The only way to convert a string of  $i$  characters into the empty string is with  $i$  deletions, represented as

a.  $E(0,j) = j$

b.  $E(i,j) = 1$

c.  $E(0,i) = j$

d.  $E(i,0) = i$  Page - 78

74. Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in an

a. Array

b. Table Page - 86

c. Variable

d. class

75. We can use the optimal substructure property to devise a formulation of the edit distance problem.

- a. Selective
- b. Optimum
- c. Iterative

d. **Recursive** Page - 78

76. Sorting is performed on the basis of \_\_\_\_\_.

- a. Computational resources
- b. Asymptotic notation
- c. Summation

d. **Some key value of attribute** page- 39

77. In Heap Sort algorithm, we call Build-heap procedure \_\_\_\_\_.

- a. **Only once** page 46
- b. Twice
- c. Thrice
- d. As many times as we need

78. Radix sort is not a non-comparative integer sorting algorithm.

- a. **True** Google Search
- b. False

79. In the statement “output P[1].x, P[1].y”, the number of times elements of P are accessed is \_\_\_\_\_.

- a. 1
- b. **2** page 14

c. 3

d. 4

80. The main purpose of mathematical analysis is measuring the \_\_\_\_\_ required by the algorithm.

a. Space

**b. Execution time**

**P-13**

c. Inputs & outputs

d. Execution time and memory

81. \_\_\_\_\_ provides us more accurate result when input values are not closer with each other

a. Average

**Median (p-34)**

c. Mode

d. Mean

82. The process of \_\_\_\_\_ ends when you are left with such tiny pieces remaining that it is trivial to solve them.

a. Brute-force

b. Plan-sweep

**c. Divide and Conquer**

**P-27**

d. Axis-sweep

83. \_\_\_\_\_ overcomes the limitations of \_\_\_\_\_ by working as per positional notations of numbers.

a. Counting sort, Radix sort

**b. Radix sort, Counting sort**

**P-71**

84. Memorization is a part of Dynamic Programming strategy.

a. True

P-74

b. False

85. Rank of an element can be defined as \_\_\_\_\_.

a. One minus the number of elements that are smaller

b. Two plus the number of elements that are greater

One plus the number of elements that are smaller P-34

c. Two minus the number of elements that are smaller

86. If the time complexity of an algorithm is given by  $O(1)$ , then its time complexity would be

a. Polynomial

b. Exponential

c. Constant - Wikipedia

d. Average

87. Quick sort is a recursive algorithm.

a. True Wikipedia ; Google

b. False

88. The asymptotic growth of  $n(n+1)/2$  is:

$O(n^2)$  As the  $n^2$  term has the largest contribution, the Big-O complexity is  $O(n^2)$

a.  $O(n)$

b.  $O(n+2)$

c.  $O(n \log n)$

89. Approach of solving geometric problems by sweeping a line across the plane is called \_\_\_\_\_ sweep.

- a. Line
- b. **Plane** **Page 18**
- c. Cube
- d. Box

90. As per algorithm of Dynamic Programming, we need to store

- a. First sub-problem only
- b. Best solution only
- c. **Intermediate sub-problems** **Pg:75**
- d. Final solution only

91. In Sieve technique, we solve the problem

- a. **In recursive manner** **Pg:34**
- b. Non recursively
- c. Using Merge Sort algorithm
- d. Using Brute force technique

92. One of the limitation in 0/1 knapsack is that an item can either be \_\_\_\_\_ in the bag or not.

- a. Use
- b. **Put** **Pg:91**
- c. Move
- d. Store

93. Which one is not passed as parameter in Quick sort algorithm?

a. End of the array

b. Middle of the array

c. Array (containing input elements) **Google**

d. Start of the array

94. In the analysis of Selection algorithm, we get the convergent \_\_\_\_\_

a. Harmonic

b. Linear

c. Arithmetic

d. **Geometric** **Pg:37**

95. A Random Access Machine (RAM) is an idealized machine with random access memory.

a. **Infinite large** **Pg:10**

b. 512 MB

c. 256 MB

d. 2 GBs

96. While analyzing Selection algorithm, we make a number of passes, in fact it could be as many as

a.  $n(n+1)$

b.  **$\log(n)$**  **Pg:37**

c.  $n/3$

d.  $n/4$

97. In Random Access Machine (RAM), instructions are executed in

a. Parallel

b. Batch

c. One by One

Pg:10

d. Multiple times

98. In selection problem, the rank of an element will be its \_\_\_\_\_ position

a. First

b. final

Pg:34

c. Second last

d. Last

99. The worst-case running time of Merge sort is \_\_\_\_\_ in order to sort an array of  $n$  elements.

a.  $O(\log n)$

b.  $O(n)$

c.  $O(n \log n)$

page 40 and google

d.  $O(n)$

100.  $f(n)$  and  $g(n)$  are asymptotically equivalent. This means that they have essentially the same \_\_\_\_\_.

a. Results

b. Variables

c. Size

d. Growth rates

P:23

101. An algorithm is a mathematical entity. Which is independent of \_\_\_\_\_.

a. Programming language

b. Machine and Programming language

c. Compiler and Programming language

d. Programming language Compiler and Machine

P:  
07

102. In Quick sort algorithm, Pivots form \_\_\_\_

- a. Stack
- b. Queue

c. Binary Search Tree

P:49

d. Graph

103. Counting sort is suitable for sorting the elements within range 1 to P. where

a. P is large

b. P is small

P-57

c. P is very large

d. P is undetermined

104. In asymptotical analysis of  $n^{(5/2)-3}$ , as n becomes large, the dominant (fastest growing) term is some constant times

- a.  $n_1$
- b. n
- c.  $n+1$

d.  $n*n$

P-23

105. \_\_\_\_ Items are not allowed in the 0/1 knapsack. a. Lighter

b. Fractional

P-91

c. Whole

d. Weighty

106. Fibonacci Sequence was named on \_\_\_\_\_, a famous mathematician in 12th Century.

- a. Fred Brooks
- b. Grady Booch

**c. Leonardo Pisano**

**P-73**

- d. Edgar F. Codd

107. In Heap Sort algorithm, we build \_\_\_\_\_ for ascending sort.

**a. Max heap**

**P-41**

- b. Min heap

108. Bubble sort is not an in-place sorting algorithm.

- a. True

**b. False**

**P-54**

109. In partition algorithm, the subarray \_\_\_\_\_ has elements which are greater than pivot element x.

- a.  $A[p..r]$
- b.  $A[p..q-1]$
- c.  $A[q]$

**d.  $A[q+1..r]$**

**P-46**

110. In Heap Sort algorithm, if heap property is violated

**a. We call Build heap procedure**

**P-43**

- b. We call Heapify procedure
- c. We ignore

d. Heap property can never be violated

111. \_\_\_\_\_ is not a characteristic of Random Access Machine.

a. Single-Processor

P-10

b. Assigning a value to a variable

c. Locality of reference

d. Executing an arithmetic instruction

112. The only way to convert an empty string into a string of  $j$  characters is by doing  $j$  insertions, represented as \_\_\_\_\_

a.  $E(i,j) = 1$

b.  $E(l,0) = l$

c.  $E(0,j) = j$

page 78

d.  $E(1,j) = j$

113. In Selection problem, the Sieve technique works in \_\_\_\_\_.
- a. Non-recursive manner
  - b. Constant time
  - c. Phases page 34
  - d. One complete go
114. Algorithm is a sequence of computational steps that ---- the input into output.
- a. Merge
  - b. Assign
  - c. Transform page 7
  - d. Integrate
115. If  $p_j$  dominates  $p_i$  and  $p_i$  dominates  $p_h$  then  $p_j$  also dominates  $p_h$ , it means dominance relation is
- a. Transitive page 18
  - b. Non Transitive
  - c. Equation
  - d. Symbolic
116. To find maximal points in brute-force algorithm each point of the space is compared against \_\_\_\_\_ of that space.
- a. One other point
  - b. All other points page 11
  - c. Few other points
  - d. Most of the other points

117. In the following code the statement "cout<<j;"executes ----  
----- times. for (j=1; j<=5; j = j+2)

```
cout<<j;
```

a. 5 times

b. 2 times

**c. 3 times**

d. 0 times

118. In merge sort algorithm, we split the array around the \_\_\_\_\_ index q. a. Entering

e. b. Mid page 17

c. Exiting

d. Summing

119. In Selection problem, the Sieve technique \_\_\_\_\_.

a. Add some more input items each time

b. Do not work recursively

c. Do not uses Divide and Conquer approach

f. Eliminates undesired data items each time

120. Consider three matrices X, Y, Z of dimensions 1 x 2, 2 x 3, 3 x 4 respectively. The number of multiplications of X(YZ) is .

a. 16

b. 32

c. 26

g. d. 32 page 84

121. In Heap Sort algorithm, the total running time for Heapify procedure is

\_\_\_\_\_

a. Theta ( $\log n$ )

b. Order ( $\log n$ )

c. Omega ( $\log n$ )

h.  $O(1)$  i.e. Constant time

122. The sieve technique works where we have to find \_\_\_\_\_ items(s) from a large input.

a. **Single** page 34

Two

b. Three

c. Similar

123. In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object  $i$ , then we gain\_\_\_\_\_

a.  $W$ (Total Weight of Knapsack)

b.  $V$  (Total Value of all items)

c.  $v_i$  (Value of object  $i$ ) page 93

d. None of the given option

124. While Sorting, the order domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.

a.  $x < y$  page 39

b.  $x > y$

c.  $x = y$

d. All of the above

125. For solving Selection problem, we introduced Sieve technique due to \_\_\_\_\_

a. **Using Decrease and Conquer strategy** page 34

b. Avoiding to sort all input data

c. Eliminating Rank of an element

d. Using Brute-force approach

126. \_\_\_\_\_ is one of the few problems, where provable lower bounds exist on how fast we can sort.

a. Searching

b. **Sorting** page 38

c. Both Searching & sorting

d. Growing

127. In plane sweep approach, a vertical line is swept across the 2d-plane from \_\_\_\_\_.

a. Right to Left

b. **Left to Right** page 18

c. Top to Bottom

d. Bottom to top

128. **In generating** Fibonacci sequence, we can avoid unnecessary repetitions by \_\_\_\_\_ process.

a. Tokenization

b. **Memorization** page 43

c. Randomization

d. Memorization

129. For \_\_\_\_\_ values of n, any algorithm is fast enough.

a. **Small** page 14

b. Medium

c. Large

d. Infinity

130. Dynamic programming comprises of \_\_\_\_\_.
- a. Recursion only
  - b. Repetition only
  - c. Recursion with Repetition
  - d. **No Repetition but Recursion** **page 75**
131. The function  $f(n) = n(\log n + 1)/2$  is asymptotically equal to  $n \log n$ : Here Lower Bound means function  $f(n)$  grows asymptotically at \_\_\_ as fast as  $n \log n$ .
- a. **Least** **page 23**
  - b. Normal
  - c. Most
  - d. At
132. Counting sort has time complexity.
- a.  $O(n+k)$
  - b.  **$O(n)$**  **page 58**
  - c.  $O(k)$
  - d.  $O(n \log n)$
133. Due to left complete nature of binary tree, the heap can be stored in
- a. **Array** **page 40**
  - b. Structures
  - c. Link List
  - d. Stack
134. Single item from a larger set of \_\_\_\_\_.
- a. Constant

**b. Pointers**

**c. Phases**

**d. n items** page 34

**135. In the clique cover problem, for two vertices to be in the same group, they must be \_\_\_\_\_ each other.**

**a. Apart from**

**b. Far from**

**c. Near to**

**d. Adjacent to** page 76

136. How much time merge sort takes for an array of numbers?

a.  $T(n^2)$

b.  $T(n)$

c.  $T(\log n)$

d.  $T(n \log n)$  page 40

137. In in-place sorting algorithm is one that uses arrays for storage.

a. No additional array page 54

b. An additional array

c. Both of above may be true according to algorithm

d. More than 3 arrays of one dimension

138. Brute-force algorithm for 2D-Maxima is operated by comparing \_\_\_\_\_ pairs of points.

a. Two

b. Some

c. Most

d. All page 18

139. While Sorting, the ordered domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.

a.  $x > y$

b.  $x < y$

c.  $x = y$

d. All of the above page 38

140. Quick sort is.

a. Stable & in place

b. Not stable but in place page 54

c. Stable but not in place

d. Some time stable & some times in place

141. Which may be a stable sort?

a. Merger

b. Insertion

c. Both above page 54

d. None of the above

142. For the Sieve Technique we take time.

a.  $T(nk)$  page 34

b.  $IT(n / 3)$

c.  $n^2$

d.  $n/3$

143. Continuation sort is suitable to sort the elements in range 1 to k.

a. K is Large

b. K is not known

c. K may be small or large

d. K is small page 54

144. Asymptotic growth rate of the function is taken over \_\_\_\_\_ case running time. .

a. Best

b. Worst page 14

c. Average

d. Normal

145. The sieve technique is a special case, where the number of sub problems is just.

a. 5

b. Many

c. 1 page 34

d. Few

146. In Quick sort, we don't have the control over the sizes of recursive calls.

**a. True** page 49

**b. False**

**c. Less information to decide**

**d. Either true or false**

147. Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their \_\_\_\_\_ coordinates. .

**a. X** page 18

**b. Y**

**c. Z**

**d. X , Y**

148. Random access machine or RAM is a/an.
- a. Machine build by Al-Khwarizmi
  - b. Mechanical machine
  - c. **Mathematical model** page 10
  - d. Electronics machine
149. The Huffman codes provide a method of encoding data inefficiently when coded using ASCII standard.
- a. True
  - b. **False** page 99
150. A heap is a left-complete binary tree that confirms to the \_\_\_\_\_.
- a. increasing order only
  - b. decreasing order only
  - c. **heap order** page 40
  - d. log n order
151. If we associate (x, y) integers pair to cars where x is the speed of the car and y is the negation of the price. High y value for a car means a \_\_\_\_\_ car.
- a. Fast
  - b. Slow
  - c. Expensive
  - d. **Cheap**
152. Which one of the following sorting algorithms is the fastest?
- a. Merge sort
  - b. **Quick sort**
  - c. Insertion sort
  - d. Heap sort

153. Quick sort algorithm divide the entire array into \_\_\_\_\_ sub arrays.

**2**

a. 3

b. 4

c. 5

154. In brute force algorithm, we measure running time  $T(n)$  based on \_\_\_\_\_.

a. Average-case time and best-case time

b. **Worst-case time and average-case time** page 46

c. Worst-case time and best-case time

d. Best-case time and starting-case time

155. For 2D Maxima problem. Plane Sweep algorithm first of all \_\_\_\_\_.

- a. Sorts all points
- b. Delete some points
- c. Output the elements
- d. Pushes all points on stack

156. There are \_\_\_\_\_ entries in the Edit Distance Matrix

- a.  $\Theta(n)$
- b.  $\Theta(n^2)$  page 84
- c.  $\Theta(n+2)$
- d.  $\Theta(n + 100)$

157. Which symbol is used for Omega notation?

- a. (O)
- b. (e)
- c. ( $\Omega$ )
- c. (@)

158. Selection sort is a \_\_\_\_\_ sorting algorithm

- d. a. In-place page 54
- b. Not In-Place
- c. Stable
- d. in-partition

159. In Dynamic Programming based solution of knapsack problem, to compute entries of 'V', we will imply a(n) \_\_\_\_\_ approach.

- a. Subjective
- b. Inductive

- c. Brute Force
- d. Combination

160. We do not need to prove comparison-based sorting algorithms by mathematically. It always takes \_\_\_\_\_ time.

- a. Big Oh  $n \log(n)$
- b. Omega  $n \log(n)$
- c. Omega  $n(n^2)$
- d. Theta  $n \log(n)$

161. Merge sort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm

- a. Not in-place, not stable one
- b. In-place, not stable one
- c. In-place, stable one

d. **Not in-place, stable one** page 54

162. Cubic function will \_\_\_\_\_ a quadratic function.

- a. Prove
- b. be equal to
- c. **overtake** Page 25
- d. find

163. Insertion sort is a \_\_\_\_\_ sorting algorithm

- a. Unstable
- b. **In-place** Page 54
- c. Not In-Place
- d. in-partition

164. To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is \_\_\_\_\_.
- Big-oh notation
  - Theta notation
  - Omega notation
  - All of the given
165. Asymptotic growth of  $8n^2 + 2n - 3$  is:
- $\Theta(n^2 + n)$
  - $\Theta(n^2)$  page 14
  - $\Theta(8n^2)$
  - $\Theta(8n^2 + 2n)$
166. In the analysis of algorithms, \_\_\_\_\_ plays an important role.
- text analysis
  - time
  - growth rate
  - money

167. In inductive approach of knapsack problem, we consider 2 cases, \_\_\_\_\_

Or \_\_\_\_\_.

a. Median, Mode

b. Recursive, Iterative

c. Leave object, Take object page 93

d. Sequentially. Parallel

168. Random Access Machine (RAM) can execute \_\_\_\_\_ instructions

a. only logical

b. parallel

c. only arithmetic

d. logical and arithmetic

169. Using \_\_\_\_\_ algorithm, efficiency is not given much importance

a. Greedy

b. Merge sort

c. Processing as there is no algorithm by this name

d. Brute Force

170. Bubble sort takes theta \_\_\_\_\_ in the worst case

a.  $(n^2)$  page 39

b.  $(n)$

c.  $\log(n)$

d.  $n\log(n)$

171. If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then dimension of resultant matrix is:

a.  $p \times q$

b.  $p \times r$  page 84

c.  $q \times r$

d.  $r \times p$

172. Dynamic Programming algorithms often use some kind of \_\_\_\_\_ to store the results of intermediate sub-problems

a. table (Page 75)

b. variable

c. stack

d. loop

173. \_\_\_\_\_ is in-place sorting algorithm.

**a. Bubble sort (Page 54)**

b. Merge sort

c. Linear search

d. Binary Search

174. Which one of the following problems can be solved using dynamic problem?

a. Bubble sort problem

**b. Matrix chain multiplication problem page 85**

c. Greedy search problem

d. Fractional knapsack problem

175. In chain matrix multiplication, solutions of the sub-problems are stored in a

\_\_\_\_\_.

a. Array

**b. Table page 86**

c. Tree

d. Link list

176. What is the average running time of a quick sort algorithm?

a.  $O(n^2)$

b.  $O(n)$

**c.  $O(n \log n)$  (Page 49)**

d.  $O(\log n)$

177. Sorting Algorithms having O \_\_\_\_\_ running time are considered to be slow ones.

a. (n)

b. ( $n^2$ ) (Page 39)

c. ( $n\log(n)$ )

d. ( $\log(n)$ )

178. While solving Selection problem, in Sieve technique we partition input data

\_\_\_\_\_

a. In increasing order

b. In decreasing order

c. According to Pivot

d. Randomly

179. \_\_\_\_\_ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.

a. Loop

b. Memoization page 74

c. Recursion

d. Function

180. In average-case time the probability of seeing input is denoted by \_\_\_\_\_.

a.  $p\{I\}$

b.  $p[I]$

c.  $p\langle i \rangle$

d.  $p(i)$  page 13

181. While applying the Sieve technique to selection sort, how to choose a pivot element.

a. Through mean

b. Linear

c. Randomly page 35

d. Sequentially

182. Number of \_\_\_\_\_ of the pseudo code are counted to measure the running time.

a. Inputs

b. Outputs

c. Steps page 13

d. Pages

183. Developing a dynamic programming algorithm generally involves \_\_\_\_\_ separate steps.

a. One

b. Two page 75

c. Three

d. Four

184.  $8n^2+2n+3$  will exceed  $c28(n)$ , no matter how large we make \_\_\_\_\_.

a. n

b. 2n

c. c2 page 25

d. this quadratic equation

185. The running time of quick sort algorithm\_\_\_\_\_.
- a. Is impossible to compute
  - b. Has nothing to do with pivot selection
  - c. Is Random upon each execution
  - d. **Greatly influenced by the selection of pivot** **page 49**
186. \_\_\_\_\_ involves breaking up the problem into sub problems whose solutions can be combined to solve the global problem.
- a. Complexity Theory
  - b. Dynamic programming solution
  - c. **Divide and Conquer Strategy** **page 34**
  - d. Greedy Algorithms
187. In \_\_\_\_\_ we have to find rank of an element from given input.
- a. Merge sort algorithm
  - b. **Selection problem** **page 34**
  - c. Brute force technique
  - d. Plane Sweep algorithm
188. How many steps are involved to design the dynamic programming strategy?
- a. 2
  - b. 3
  - c. 1
  - d. **4** **page 92**
189. In Bucket sort, if there are duplicates then each bin can be replaced by a a. Stack

**b. Linked list** page 69

c. Hash table

d. Heap

190. In merge sort algorithm, we split the array \_\_\_\_\_ to find index q.

a. from start

**b. midway** page 28

c. from end

d. both from start or end

191. Find the maximum value of the items which can carry using knapsack Knapsack weight capacity = 50.

Item Weight Value

11070,22020,33080,470 200

a. 280

b. 100

c. 90

**200**

192. In 2-d maxima problem a point p is said to be dominated by point q if

\_\_\_\_\_.

a.  $p.x \leq q.x$

**b.  $p.x \leq q.x$  and  $p.y \leq q.y$**  page 17

c.  $p.y \leq q.y$

d.  $p.x \geq q.x$  and  $p.y \geq q.y$

193. Sorting can be in \_\_\_\_\_.

- a. Increasing order only
- b. Decreasing order only
- c. Both increasing and decreasing order
- d. Random order

194. Recurrence can be described in terms of \_\_\_\_\_.

- a. Array
- b. Linear
- c. Tree page 31
- d. Graph

195. The brute-force algorithm for 2D-Maxima runs in order  $O(\_)$  time.

- a.  $n$
- b.  $n(\log n)$
- c.  $n*n$  page 18
- d.  $n^3$

196. In plane sweep approach of solving geometric problems, a \_\_\_\_\_ is swept across the plane.

- a. Line page 18
- b. Plane
- c. Cube
- d. Box

197. Which of the following is calculated with Big Omega notation?

- a. Medium bounds
- b. Upper bounds

**c. Lower bounds** Page - 25

d. Both upper and lower bounds

198. \_\_\_\_\_ is always based on divide and conquer strategy.

a. Bubble sort

b. Selection sort

c. Pigeon sort

**d. Quick sort** page 46

199. If a matrix has three rows and two columns, then dimensions of matrix will be:

a. 3x2

**2x3**

b. 3x3

c. 2x2

200. Asymptotic notations are used to describe \_\_\_\_\_ of an algorithm.

a. Length

**b. running time** google

c. size

d. compile time

201. Catalan numbers are related the number of different \_\_\_\_\_ on 'n' nodes.

a. Arrays

b. linked lists

**c. binary trees** page 85

d. functions

202. Applying the sieve technique to selection problem, \_\_\_\_\_ element is picked from array.

- a. Output
- b. Total
- c. Input

**d. Pivot** **page 35**

203. Dynamic Programming approach is usually useful in solving \_\_\_\_\_ problems.

a. Normal

**b. Optimization** **google**

- c. Array
- d. Loop

204. In recursive formulation of knapsack Problem:  $V [0, j] =$  \_\_\_\_\_ for  $j \geq 0$

a. -1

**b. 0** **page 93**

- c. 1
- d. 2

205. \_\_\_\_\_ is a linear time sorting algorithm.

a. Merge sort

**b. Radix sort** **page 71**

c. Quick sort

d. Bubble sort

206. Quick sort is one of the \_\_\_\_\_ sorting algorithm.

a. **Fastest** page 19

b. Slowest

c. Major

d. Average

207. The time assumed for each basic operation to execute on RAM model of computation is \_\_\_\_\_.

a. Infinite

b. Continuous

c. **Constant** page 10

d. Variable

208. In Sieve Technique, we know the item of interest.
- True
  - False** page 34
209. While analyzing algorithms, \_\_\_\_\_ and \_\_\_\_\_ usually considered difficult to calculate.
- Finite, Infinite
  - Floor, ceiling** google
  - Row, Column
  - Graph, Tree
210. While analysis of the brute-force maxima algorithm, an array sorted in the reverse order is the type of \_\_\_\_\_ case input.
- Best
  - Worst** page 14
  - Somewhat bad
  - Average
211. \_\_\_\_\_ is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.
- Mean
  - Mode
  - Average
  - Median** page 34

212. In asymptotical analysis of  $n(n-3)$  and  $4n*n$ , as  $n$  becomes large, the dominant (fastest growing) term is some constant times \_\_\_\_\_.
- a.  $n+1$
  - b.  $n-1$
  - c.  $n$
  - d.  $n*n$  page 23
213. In addition to passing in the array itself to Merge Sort algorithm, we will pass in other arguments which are indices.

a. Two

P-38

b. Three

c. Four

d. Five

214. In 2d-maximal problem, a point is said to be if it is not dominated by any other point in that space.

a. Member

b. Minimal

**c. Maximal**

**P-11**

d. Joint

215. Counting sort assumes that the numbers to be sorted are in the range \_\_\_\_\_.

a. K to n where n is large

**b. 1 to k where k is small**

**(P-57)**

c. K to n where k is small

d. k to n where n is small

216. Insertion sort is an efficient algorithm for sorting a \_\_\_\_\_ number of elements

**a. Large**

**P-39**

b. Small

c. Extra large

d. Medium

217. If the indices passed to merge sort algorithm are \_\_\_\_\_ then this means that there is only one element to sort.

**a. Small page 28**

b. Large

c. Equal

d. Not Equal

218. In Knapsack Problem, each item must be entirely accepted or rejected, is called \_\_\_\_\_ problem.

a. Fractional

**b. 0-1 P-92**

c. Linear

d. Optimal

219. If the time complexity of an algorithm is  $O(n)$ . then it is called \_\_\_\_\_ time complexity.

a. **Linear** [Wikipedia](#)

b. Constant

c. Average

d. Exponential

220. In the case of \_\_\_\_\_ analysis does not depend upon on the distribution of input.

a. Merge sort

**b. Quick sort** [P-50](#)

c. Insertion sort

d. Heap sort

221. We can use the \_\_\_\_\_ Property to devise a recursive formulation of the edit distance problem.

a. Small substructure

b. Algorithmic

c. Real

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

**CS502 GRAND QUIZ SOLVED**

**Prepared by VUAnswer.com**

1. The sequence of merge sort algorithm is:
  - a. Divide Combine-Conquer
  - b. Conquer-Divide-Combine
  - c. Divide-Conquer-Combine Page 27**
  - d. Combine-Divide-Conquer
2. In \_\_\_\_\_ Knapsack Problem, limitation is that an item can either be put in the bag or not. Fractional items are not allowed.
  - a. 0
  - b. 1
  - c. 0/1 Page 91**
  - d. Fractional
3. In Selection algorithm, we assume pivot selection takes theta \_\_\_\_\_ runn  
gtime.
  - a. n Page - 36**
  - b.  $n^2$
  - c.  $n^3$
  - d.  $\log(n)$
4. In Heap Sort algorithm (using max heap), when every time maximum

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

elements removed from top \_\_\_\_.

- a. We call merge Sort Algorithm
- b. it becomes Order  $n^2$  Algorithm
- c. Divide and Conquer strategy helps us
- d. **We are left with a hole** **Page – 41**

5. If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then each entry in resultant matrix takes \_\_\_\_\_ time.

- a.  **$O(q)$**  **Page - 84**
- b.  $O(1)$
- c.  $O(p \times q)$
- d.  $O(q \times r)$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

6. \_\_\_\_\_ is a method of solving a problem in which we check all possible solutions to the problem to find the solution we need.
- Plane-Sweep Algorithm
  - Sorting Algorithm
  - Brute-Force Algorithm** google
  - Greedy approach
7. The worst case running time of Quick sort algorithm \_\_.
- Cannot be quadratic
  - Is quadratic** google [QuickSort and its Analysis \(codesdope.com\) & http://personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Sorting/quickSort.htm](http://personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Sorting/quickSort.htm)
  - Is always Exponential
  - Is linear
8. In max heap (for Heap Sort algorithm), when every time maximum element is removed from top we replace it with \_\_ leaf in the tree.
- second last
  - Last** Page -41
  - First
  - Any
9. Quick sort algorithm was developed by -
- AlferdAho
  - Sedgewick
  - John Vincent Atanasoff
  - Tony Hoare** – Google wikipedia
10. If Matrix-A has dimensions "3x2" and Matrix-B has dimensions "2x3", then multiplication of Matrix-A and Matrix-B will result a new Matrix-C having dimensions.

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

a.  $3 \times 2$

b.  $2 \times 3$

c.  $2 \times 2$

d.  **$3 \times 3$**  <http://www.calcul.com/show/calculator/matrix-multiplication>

11. For comparison-based sorting algorithms, it is possible to sort more efficiently than  $\Omega(n \log(n))$  time.

a. Always

b. **Not** **P-54**

c. Sometimes

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

- d. Sometimes not
12. Dynamic Programming approach is usually useful in solving optimization problems.
- a. **True** [https://en.wikipedia.org/wiki/Dynamic\\_programming](https://en.wikipedia.org/wiki/Dynamic_programming) & <http://web.mit.edu/15.053/www/AMP-Chapter-11.pdf>
- b. False
13. In Sorting the key value or attribute \_\_\_ from an ordered domain.
- a. **Must be** **page 39**
- b. Not always
- c. May be
- d. Occasionally
14. Result of asymptotical analysis of  $n(n-3)$  and  $4n*n$  is that \_\_\_
- a.  $n(n-1)$  is asymptotically Less
- b.  $n(n-1)$  is asymptotically Greater
- c. Both are asymptotically Not equivalent
- d. **Both are asymptotically Equivalent** **page 23 ( $4n*n= 4n^2$ )**
15. Floor and ceiling are \_\_\_\_\_ to calculate while analyzing algorithms
- a. Very easy
- b. **Usually considered difficult** **P-31**
- c. 3rd Option is missing
- d. 4th Option is missing
16. \_\_\_\_\_ of reference is an important fact of current processor technology.
- a. Defining
- b. Assigning
- c. Formality
- d. **Locality** **P-8**

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

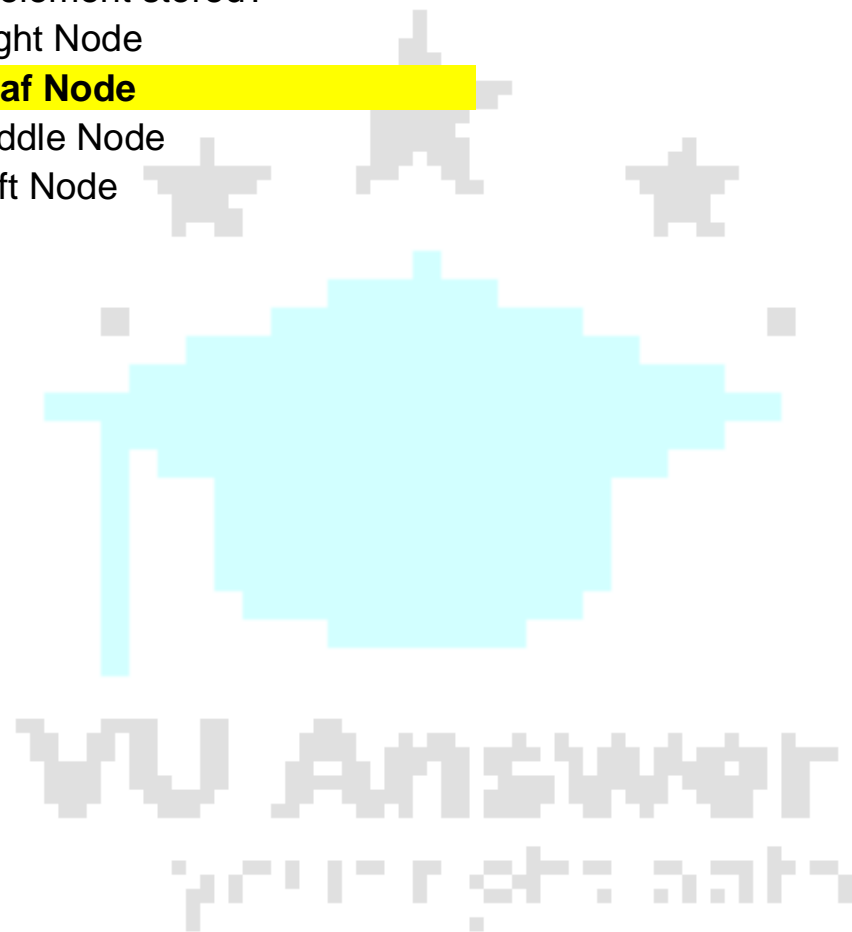
<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

17. In max-heap, largest element is stored at root node. Where is the smallest element stored?

- a. Right Node
- b. Leaf Node**
- c. Middle Node
- d. Left Node



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

18. In average-case time analysis of Quick sort algorithm, the most balanced case for partition is when we divide the list of elements into \_\_\_\_\_.
- a. **Equal no. of pieces as of input elements**
  - b. Single piece exactly
  - c. Two nearly equal pieces
  - d. Three nearly equal pieces
19. Which of the following is calculated with Big O notation?
- a. Medium bounds
  - b. **Upper bounds** **Page - 25**
  - c. Lower bounds
  - d. Both upper and lower bounds
20. Edit distance algorithm based on \_\_\_\_\_ strategy
- a. Greedy
  - b. **Dynamic Programming** **Page - 81**
  - c. Divide and Conquer
  - d. Searching
21. In Heapsort Algorithm, total time taken by heapify procedure is \_\_\_\_\_
- a.  **$O(\log n)$**  **Page-43**
  - b.  $O(\log^2 n)$
  - c.  $O(n \log n)$
  - d.  $O(n^2 \log n)$
22. Al-Khwarizmi was a/an \_\_\_\_\_
- a. Artist
  - b. **Mathematician** **P-7**
  - c. Astronomer

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

d. Khalifah

23. When matrix A of  $5 \times 3$  is multiply with metric B of  $3 \times 4$  then the number of multiplication required is: Not found exactly

a. 15

b. 12

c. 36

d. **60** Not Found exactly but as per formula at page 84,

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

# VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

24. Pseudo code of algorithms are to be read by \_\_\_\_.
- a. People** **Page -12**
  - b. RAM
  - c. Computer
  - d. Compiler
25. The sieve technique is a special case, where the number of sub-problems is Just
- 
- a. 1** **P-34**
  - b. 2
  - c. 3
  - d. 4
26. When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has \_sub-problems.
- a. Overlapping** **– Google Search**
  - b. Over costing
  - c. Optimized
  - d. Three
27. Sieve technique is very important special case of Divide-and-Conquer strategy.
- a. True** **P-34**
  - b. False
28. In order to say anything meaningful about our algorithms, it will be important for us to settle on a\_\_.
- a. Java Program
  - b. C++ Program
  - c. Pseudo program

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

**d. Mathematical model of computation P-10**

29. Merge sort is based on\_\_\_\_\_.

- a. Brute-force
- b. Plan-sweep
- c. Axis-sweep

**d. Divide and Conquer P-27**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

30. What time does Merge Sort algorithm take in order to sort an array of 'n' numbers?
- a.  $\Theta(n)$
  - b.  $\Theta(\log n)$
  - c.  $\Theta(n^2)$
  - d.  $\Theta(n \log n)$  Google Search**
31. In Heap Sort algorithm, the first step is to \_\_\_\_\_.
- a. Call Build-Heap procedure Page - 46**
  - b. Sort the array in descending order
  - c. Call Heapify procedure
  - d. Find the number of input elements
32. The definition of theta-notation relies on proving \_\_\_\_\_ asymptotic bound.
- a. One
  - b. Lower
  - c. Upper
  - d. Both lower & upper Page - 25**
33. In merge sort algorithm, to merge two lists of size  $n/2$  to a list of size  $n$ , takes \_\_\_\_\_ time.
- a. Theta (n) Page - 32**
  - b. Theta  $\log(n)$
  - c. Theta  $\log_2(n)$
  - d. Theta  $n \log(n)$
34. We can make \_\_\_\_\_ recursive calls in Fibonacci Sequence.
- a. Infinite
  - b. Finite google**

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Only one
  - d. Zero
35. Following is NOT the application of Edit Distance problem.
- a. Speech recognition
  - b. Spelling Correction
  - c. Ascending Sort**
  - d. Computational Molecular Biology

**Page - 76**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

36. In plane sweep approach, a vertical line is swept across the 2d-plane and structure is used for holding the maximal points lying to the left of the swepline.
- Array
  - Queue
  - Stack** **Page - 18**
  - Tree
37. When a heapify procedure is applied to the root node to restore the heap, then at each level, the comparison performed takes time:
- It will take  $(\log n)$ .
  - It can not be predicted
  - It will take  $O(1)$ .** **Page - 43**
  - Time will vary according to the nature of input data.
38. \_\_\_\_\_ time is the maximum running time over all legal inputs.
- Worst-case** **Page - 13**
  - Average-case
  - Best-case
  - Good-case
39. Efficient algorithm requires less computational...
- Memory
  - Running Time
  - Memory and Running Time** **Page - 9**
  - Energy
40. For average-case time analysis of Quick sort algorithm, Pivot selection is on average basis from \_\_\_\_\_
- half of the input values
  - all possible random values** **Page - 50**

**SUBSCRIBE OUR CHANNEL**

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

**JOIN WHATSAPP GROUP**

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Pivot is input separately
  - d. values greater than 5
41. Selection algorithm takes theta \_\_\_\_\_
- a.  $(n^2)$
  - b.  $(n)$**
  - c.  $\log(n)$
  - d.  $n \log(n)$

**Page - 37**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

42. Recurrence can be described in terms of a tree.  
**a. Yes** **Page - 31**  
b. No
43. Time complexity of Dynamic Programming based algorithm for computing the minimum cost of Chain Matrix Multiplication is \_\_\_\_  
a. Log n  
b. n  
c.  $n^2$  (n square)  
**d.  $n^3$  (n cube)** **Page -90**
44. The Iteration method is used for \_\_\_\_  
a. Comparing sorting algorithms only  
**b. Solving Recurrence relations** **Page 31**  
c. Merging elements in Merge sort  
d. Dividing elements in Merge sort
45. In 3-Dimensional space, a point P has \_\_\_\_ coordinate(s).  
a. (X, Y)  
b. (X, 0)  
c. (0, Y)  
**d. (X, Y, Z)**
46. Chain matrix multiplication problem can be solved through \_\_\_\_ strategy.  
**a. Dynamic programming** **Page - 85**  
b. Greedy  
c. Divide and conquer  
d. Sorting
47. Merge sort have running time running time of Heap sort. Not found exactly

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

a. Greater than

**b. Less than** **Google**

c. Equal to

d. Different than

48. Median is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.

a. True

**b. False** **Page – 34**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

49. We do not need to mathematically prove that for comparison-based sorting algorithms always takes Omega  $n \log(n)$  time.
- a. True **Google & VU Tech (pg 46 not very clear)**
  - b. False
50. The Omega-notation allows us to state only the asymptotic \_\_ bounds.
- a. Middle
  - b. Lower **Page 25**
  - c. Upper
  - d. Both lower & upper
51. Both lower & upper Sorting can be in \_\_\_\_\_
- a. Increasing order only
  - b. Decreasing order only
  - c. **Both Increasing and Decreasing order – (My opinion)**
  - d. Random order
52. Radix sort performs sorting the numbers \_\_\_\_\_ digit (s) at a time.
- a. One **Page - 71**
  - b. Two
  - c. Three
  - d. All
53. Quicksort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm.
- a. Not in place, not stable one
  - b. **In place , not stable one Page - 54**
  - c. In place , stable one
  - d. Not in place , stable one
54. Consider three matrices X,Y,Z of dimensions  $1 \times 2, 2 \times 3, 3 \times 4$  respectively. The number of multiplications of (XY) Z is:

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

a. **18** As per lecture slides

b. 32

c. 24

d. 30

55. In Fibonacci Sequence, unnecessary repetitions do not exist at all.

a. True

b. **False**

**Page – 74**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

56. It is not a Fibonacci sequence .      1,1,1,2,3,5,8,13,21,34,55,.....
- a. True Page - 73
  - b. False
57. Heap sort is a/ an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm.
- a. Not in place, not stable one
  - b. In place , not stable one Page - 54
  - c. In place , stable one
  - d. Not in place , stable one
58. Identify the True Statement
- a. The knapsack problem does not belong to the domain of optimization problems.
  - b. The knapsack problem belongs to the domain of optimization problems. Page - 91
  - c. The Knapsack problem cannot be solved by using dynamic programming
  - d. The knapsack problem is optimally solved by using brute force algorithm.
59. In Dynamic Programming, our approach is to \_\_\_\_\_
- a. Develop the solution in a top-down fashion
  - b. Express the problem non-recursively
  - c. Build the solution in a bottom-up fashion Page - 75
  - d. Input several sub-problems simultaneously
60. Counting sort is suitable to sort the elements in range 1 to K;
- a. K is large
  - b. K is small Page - 57
  - c. K may be large or small

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

d. None

61. We can multiply two matrices A and B only when they are compatible which means

a. **Number of columns in A must be equal to number of rows in B. it seems Correct as per page 84**

**([https://www.varsitytutors.com/hotmath/hotmath\\_help/topics/matrix-multiplication#:~:text=You%20can%20only%20multiply%20two,an%20m%C3%97p%20matrix.](https://www.varsitytutors.com/hotmath/hotmath_help/topics/matrix-multiplication#:~:text=You%20can%20only%20multiply%20two,an%20m%C3%97p%20matrix.))**

b. Number of rows and columns do not matter

**WU Answer**  
your right path

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

- c. Number of columns in A must be equal to number of columns in B  
d. Number of rows in A must be equal to number of rows in B
62. Matrix multiplication is a (n)\_\_\_\_\_ operation.  
a. Commutative  
**b. Associative** **Page 85**  
c. Neither commutative nor associative  
d. Commutative but not associative
63. In Dynamic Programming approach, solution is modified / changed  
a. Always once  
**b. At each stage** **google and wikipedia**  
c. Only for specific problems  
d. At 4<sup>th</sup> stage only
64. In Knapsack problem, the goal is to put items in the Knapsack such that the value of the items is \_\_ subject to weight limit of knapsack.  
a. Minimized  
b. Decreased  
**c. Maximized** **Page - 91**  
d. None of the given options
65. An in-place sorting algorithm is one that \_\_\_\_\_ uses additional array for storage.  
a. Always  
b. Permanently  
**c. Does not** **Page - 54**  
d. Sometime
66. Memoization is a part of Dynamic Programming Strategy.  
**a. True** **Page - 74**

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

b. False

67. If matrix A of dimension  $2 \times 4$  is multiply with matrix B of dimension  $4 \times 3$ , then the dimension of resultant matrix is Not found exactly

a.  $2 \times 4$

b.  $4 \times 3$

c.  $3 \times 4$

d.  $2 \times 3$

**It seems correct as per second last Para of page 84**

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

68. In Dynamic Programming approach, we do not store the solution to each sub-problem in case if it reappears.
- a. True
  - b. False** Page - 75
69. Dynamic Programming is a problem-solving approach in which \_\_\_\_
- a. Problem is solved in Zero time
  - b. Solution is developed only at final stage
  - c. Both are correct
  - d. Both are incorrect** google
70. In Fibonacci sequence, each term is calculated by \_\_\_\_\_ previous terms.
- 
- a. Subtracting, Two
  - b. Adding, Three
  - c. Adding, Two** Page - 73
  - d. Multiplying, Two
71. Selection sort is not an in-place sorting algorithm.
- a. True** Page - 54
  - b. False
72. If there are  $\theta(n^2)$  entries in edit distance matrix then the total running time is:
- a.  $\theta(n)$
  - b.  $\theta(1)$
  - c.  $\theta(n^2)$**  Page - 84
  - d.  $\theta(n \log n)$
73. The only way to convert a string of  $i$  characters into the empty string is with  $i$  deletions, represented as
- a.  $E(0..i) = j$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

b.  $E(i,j) = 1$

c.  $E(0,i) = j$

**d.  $E(i,0) = I$**

**Page - 78**

74. Dynamic programming formulation of the matrix chain multiplication problem will store the solutions of each sub problem in an

a. Array

**b. Table**

**Page - 86**

c. Variable

d. class

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

75. We can use the optimal substructure property to devise a formulation of the edit distance problem.
- a. Selective
  - b. Optimum
  - c. Iterative
  - d. Recursive** Page - 78
76. Sorting is performed on the basis of \_\_\_\_\_.
- a. Computational resources
  - b. Asymptotic notation
  - c. Summation
  - d. Some key value of attribute** page- 39
77. In Heap Sort algorithm, we call Build-heap procedure \_\_\_\_\_.
- a. Only once** page 46
  - b. Twice
  - c. Thrice
  - d. As many times as we need
78. Radix sort is not a non-comparative integer sorting algorithm.
- a. True** Google Search
  - b. False
79. In the statement "output P[1].x, P[1].y", the number of times elements of P are accessed is \_\_\_\_\_.
- a. 1
  - b. 2** page 14
  - c. 3
  - d. 4
80. The main purpose of mathematical analysis is measuring the

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

require

dby the algorithm.

- a. Space
- b. Execution time P-13**
- c. Inputs & outputs
- d. Execution time and memory



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

81. \_\_\_\_\_ provides us more accurate result when input values are not closer with each other
- Average
  - Median P-34**
  - Mode
  - Mean
82. The process of \_\_\_\_\_ ends when you are left with such tiny pieces remaining that it is trivial to solve them.
- Brute-force
  - Plan-sweep
  - Divide and Conquer P-27**
  - Axis-sweep
83. \_\_\_\_\_ overcomes the limitations of \_\_\_\_\_ by working as perpositional notations of numbers.
- Counting sort, Radix sort
  - Radix sort, Counting sort P-71**
84. Memorization is a part of Dynamic Programming strategy.
- True P-74**
  - False
85. Rank of an element can be defined as \_\_\_\_\_.
- One minus the number of elements that are smaller
  - Two plus the number of elements that are greater
  - One plus the number of elements that are smaller P-34**
  - Two minus the number of elements that are smaller
86. If the time complexity of an algorithm is given by  $O(1)$ , then its time complexity would be

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- a. Polynomial
- b. Exponential
- c. **Constant** - Wikipedia
- d. Average

87. Quick sort is a recursive algorithm.

- a. **True** Wikipedia ; Google
- b. False



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

88. The asymptotic growth of  $n(n+1)/2$  is:
- a.  **$O(n^2)$**  As the  $n^2$  term has the largest contribution, the Big-O complexity is  $O(n^2)$
  - b.  $O(n)$
  - c.  $O(n+2)$
  - d.  $O(n \log n)$
89. Approach of solving geometric problems by sweeping a line across the plane is called \_\_\_\_ sweep.
- a. Line
  - b. **Plane** Page 18
  - c. Cube
  - d. Box
90. As per algorithm of Dynamic Programming, we need to store
- a. First sub-problem only
  - b. Best solution only
  - c. **Intermediate sub-problems** Pg:75
  - d. Final solution only
91. In Sieve technique, we solve the problem
- a. **In recursive manner** Pg:34
  - b. Non recursively
  - c. Using Merge Sort algorithm
  - d. Using Brute force technique
92. One of the limitation in 0/1 knapsack is that an item can either be \_\_\_\_ in the bag or not.
- a. Use
  - b. **Put** Pg:91
  - c. Move

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- d. Store
93. Which one is not passed as parameter in Quick sort algorithm?
- a. End of the array
  - b. Middle of the array
  - c. **Array (containing input elements) Google**
  - d. Start of the array

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

94. In the analysis of Selection algorithm, we get the convergent \_\_\_\_\_
- a. Harmonic
  - b. Linear
  - c. Arithmetic
  - d. Geometric** **Pg:37**
95. A Random Access Machine (RAM) is an idealized machine with random access memory.
- a. Infinite large** **Pg:10**
  - b. 512 MB
  - c. 256 MB
  - d. 2 GBs
96. While analyzing Selection algorithm, we make a number of passes, in fact it could be as many as
- a.  $n(n+1)$
  - b.  $\log(n)$**  **Pg:37**
  - c.  $n/3$
  - d.  $n/4$
97. In Random Access Machine (RAM), instructions are executed in
- a. Parallel
  - b. Batch
  - c. One by One** **Pg:10**
  - d. Multiple times
98. In selection problem, the rank of an element will be its \_\_\_\_\_ position
- a. First
  - b. final** **Pg:34**
  - c. Second last

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

d. Last

99. The worst-case running time of Merge sort is \_\_ in order to sort an array of  $n$  elements.

a.  $O(\log n)$

b.  $O(n)$

**c.  $O(n \log n)$  page 40 and google**

d.  $O(n)$

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

100.  $f(n)$  and  $g(n)$  are asymptotically equivalent. This means that they have essentially the same.
- a. Results
  - b. Variables
  - c. Size
  - d. Growth rates P:23**
101. An algorithm is a mathematical entity. Which is independent of \_\_\_\_.
- a. Programming language
  - b. Machine and Programming language
  - c. Compiler and Programming language
  - d. Programming language Compiler and Machine P:07**
102. In Quick sort algorithm, Pivots form \_\_
- a. Stack
  - b. Queue
  - c. Binary Search Tree P:49**
  - d. Graph
103. Counting sort is suitable for sorting the elements within range 1 to P. where
- a. P is large
  - b. P is small P-57**
  - c. P is very large
  - d. P is undetermined
104. In asymptotical analysis of  $n^{(5^2)-3}$ , as  $n$  becomes large, the dominant (fastest growing) term is some constant times
- a.  $n_1$
  - b.  $n$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

c.  $n+1$

**d.  $n*n$  P-23**

105. \_\_\_ Items are not allowed in the 0/1 knapsack.

a. Lighter

**b. Fractional P-91**

c. Whole

d. Weighty



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

106. Fibonacci Sequence was named on \_\_\_\_, a famous mathematician in 12th Century.
- a. Fred Brooks
  - b. Grady Booch
  - c. Leonardo Pisano P-73**
  - d. Edgar F. Codd
107. In Heap Sort algorithm, we build \_\_\_\_ for ascending sort.
- a. Max heap P-41**
  - b. Min heap
108. Bubble sort is not an in-place sorting algorithm.
- a. True
  - b. False P-54**
109. In partition algorithm, the subarray \_\_\_\_ has elements which are greater than pivot element x.
- a.  $A[p..r]$
  - b.  $A[p..q-1]$
  - c.  $A[q]$
  - d.  $A[q+1..r]$  P-46**
110. In Heap Sort algorithm, if heap property is violated
- a. We call Build heap procedure P-43**
  - b. We call Heapify procedure
  - c. We ignore
  - d. Heap property can never be violated
111. \_\_\_\_ is not a characteristic of Random Access Machine.
- a. Single-Processor P-10**
  - b. Assigning a value to a variable

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Locality of reference
  - d. Executing an arithmetic instruction
112. The only way to convert an empty string into a string of  $j$  characters is by doing  $j$  insertions, represented as \_\_\_\_
- a.  $E(i,j) = 1$
  - b.  $E(l,0) = l$
  - c.  **$E(0,j) = j$**  page 78
  - d.  $E(1,j) = j$

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

113. In Selection problem, the Sieve technique works in\_\_\_\_\_.
- Non-recursive manner
  - Constant time
  - Phases page 34**
  - One complete go
114. Algorithm is a sequence of computational steps that \_\_\_\_\_ the input into output.
- Merge
  - Assign
  - Transform page 7**
  - Integrate
115. If  $p_j$  dominates  $p_i$  and  $p_i$  dominates  $p_h$  then  $p_j$  also dominates  $p_h$ , it means dominance relation is
- Transitive page 18**
  - Non Transitive
  - Equation
  - Symbolic
116. **To find maximal** points in brute-force algorithm each point of the space is compared against \_\_\_\_\_ of that space.
- One other point
  - All other points page 11**
  - Few other points
  - Most of the other points
117. In the following code the statement "cout<<j;" executes ----times.  
for (j=1; j<=5; j =

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

j+2)cout<<j;

- a. 5 times
- b. 2 times
- c. **3 times**
- d. 0 times



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

118. In merge sort algorithm, we split the array around the \_\_\_ index q.
- a. Entrin
  - gb. **Mid** page 17
  - c. Exiting
  - d. Summing
119. In Selection problem, the Sieve technique \_\_\_\_\_.
- a. Add some more input items each time
  - b. Do not work recursively
  - c. Do not uses Divide and Conquer approach
  - d. **Eliminates undesired data items each time**
120. Consider three matrices X, Y, Z of dimensions 1 x 2, 2 x 3, 3 x 4 respectively. The number of multiplications of X(YZ) is .
- a. 16
  - b. 32
  - c. 26
  - d. **32** page 84
121. In Heap Sort algorithm, the total running time for Heapify procedure is \_\_\_\_\_
- a. Theta (log n)
  - b. Order (log n)
  - c. Omega (log n)
  - d. O(1) i.e. Constant time
122. The sieve technique works where we have to find \_\_\_ items(s) from alarge input.
- a. **Single** page 34
  - b. Two

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

- c. Three
  - d. Similar
123. In Dynamic Programming based solution of Knapsack Problem, if we decide to take an object  $i$ , then we gain \_\_\_
- a.  $W$  (Total Weight of Knapsack)
  - b.  $V$  (Total Value of all items)
  - c.  $v_i$  (Value of object  $i$ ) **page 93**
  - d. None of the given option

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

124. While Sorting, the order domain means for any two input elements  $x$  and  $y$  \_\_\_\_\_ satisfies only.
- a.  $x < y$  page 39
  - b.  $x > y$
  - c.  $x = y$
  - d. All of the above
125. For solving Selection problem, we introduced Sieve technique due to \_\_\_\_\_
- a. Using Decrease and Conquer strategy page 34
  - b. Avoiding to sort all input data
  - c. Eliminating Rank of an element
  - d. Using Brute-force approach
126. \_\_\_\_\_ is one of the few problems, where provable lower bounds exist on how fast we can sort.
- a. Searching
  - b. Sorting page 38
  - c. Both Searching & sorting
  - d. Growing
127. In plane sweep approach, a vertical line is swept across the 2d-plane from \_\_\_\_\_.
- a. Right to Left
  - b. Left to Right page 18
  - c. Top to Bottom
  - d. Bottom to top
128. In generating Fibonacci sequence, we can avoid unnecessary repetitions by

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

\_\_\_\_\_ process.

- a. Tokenization
- b. Memorization page 43**
- c. Randomization
- d. Memorization

129. For \_\_\_\_\_ values of n, any algorithm is fast enough.

- a. Small page 14**
- b. Medium
- c. Large
- d. Infinity

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

130. Dynamic programming comprises of \_\_\_\_\_.  
a. Recursion only  
b. Repetition only  
c. Recursion with Repetition  
**d. No Repetition but Recursion** **page 75**
131. The function  $f(n)=n(\log n+1)/2$  is asymptotically equal to  $n \log n$ . Here Lower Bound means function  $f(n)$  grows asymptotically at \_\_\_ as fast as  $n \log n$ .  
**a. Least** **page 23**  
b. Normal  
c. Most  
d. At
132. Counting sort has time complexity.  
a.  $O(n+k)$   
**b.  $O(n)$**  **page 58**  
c.  $O(k)$   
d.  $O(n \log n)$
133. Due to left complete nature of binary tree, the heap can be stored in  
**a. Array** **page 40**  
b. Structures  
c. Link List  
d. Stack
134. Single item from a larger set of \_\_\_\_\_.  
a. Constant  
b. Pointers  
c. Phases  
**d. n items** **page 34**

**SUBSCRIBE OUR CHANNEL**

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

**JOIN WHATSAPP GROUP**

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

135. In the clique cover problem, for two vertices to be in the same group, they must be \_\_\_\_\_ each other.
- a. Apart from
  - b. Far from
  - c. Near to
  - d. Adjacent to**

page 76

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

136. How much time merge sort takes for an array of numbers?
- a.  $T(n^2)$
  - b.  $T(n)$
  - c.  $T(\log n)$
  - d.  $T(n \log n)$  page 40
137. In in-place sorting algorithm is one that uses arrays for storage.
- a. No additional array page 54
  - b. An additional array
  - c. Both of above may be true according to algorithm
  - d. More than 3 arrays of one dimension
138. Brute-force algorithm for 2D-Maxima is operated by comparing \_\_\_ pairs of points.
- a. Two
  - b. Some
  - c. Most
  - d. All page 18
139. While Sorting, the ordered domain means for any two input elements x and y \_\_\_ satisfies only.
- a.  $x > y$
  - b.  $x < y$
  - c.  $x = y$
  - d. All of the above page 38
140. Quick sort is.
- a. Stable & in place
  - b. Not stable but in place page 54
  - c. Stable but not in place

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- d. Some time stable & some times in place
141. Which may be a stable sort?
- a. Merger
  - b. Insertion
  - c. **Both above** **page 54**
  - d. None of the above

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

142. For the Sieve Technique we take time.
- a.  $T(nk)$  page 34
  - b.  $IT(n / 3)$
  - c.  $n^2$
  - d.  $n/3$
143. Continuation sort is suitable to sort the elements in range 1 to k.
- a. K is Large
  - b. K is not known
  - c. K may be small or large
  - d. K is small page 54
144. Asymptotic growth rate of the function is taken over \_\_\_ case runningtime. .
- a. Best
  - b. Worst page 14
  - c. Average
  - d. Normal
145. The sieve technique is a special case, where the number of sub problems is just.
- a. 5
  - b. Many
  - c. 1 page 34
  - d. Few
146. In Quick sort, we don't have the control over the sizes of recursive calls.
- a. True page 49
  - b. False

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Less information to decide
  - d. Either true or false
147. Before sweeping a vertical line in plane sweep approach, in start sorting of the points is done in increasing order of their coordinates. .
- a. **X** **page 18**
  - b. Y
  - c. Z
  - d. X, Y

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

148. Random access machine or RAM is a/an.
- a. Machine build by Al-Khwarizmi
  - b. Mechanical machine
  - c. Mathematical model** page 10
  - d. Electronics machine
149. The Huffman codes provide a method of encoding data inefficiently whencoded using ASCII standard.
- a. True
  - b. False** page 99
150. A heap is a left-complete binary tree that confirms to the \_\_\_\_.
- a. increasing order only
  - b. decreasing order only
  - c. heap order** page 40
  - d. log n order
151. If we associate (x, y) integers pair to cars where x is the speed of the car and y is the negation of the price. High y value for a car means a \_\_\_\_ car.
- a. Fast
  - b. Slow
  - c. Expensive
  - d. Cheap**
152. Which one of the following sorting algorithms is the fastest?
- a. Merge sort
  - b. Quick sort
  - c. Insertion sort
  - d. Heap sort

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

153. Quick sort algorithm divide the entire array into \_\_\_\_\_ sub arrays.
- 2**
  - 3
  - 4
  - 5
154. In brute force algorithm, we measure running time  $T(n)$  based on \_\_\_\_.
- Average-case time and best-case time
  - Worst-case time and average-case time** **page 46**
  - Worst-case time and best-case time
  - Best-case time and starting-case time

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

155. For 2D Maxima problem. Plane Sweep algorithm first of all\_\_\_\_\_.
- Sorts all points**
  - Delete some points
  - Output the elements
  - Pushes all points on stack
156. There are\_\_\_\_\_ entries in the Edit Distance Matrix
- $\Theta(n)$
  - $\Theta(n^2)$  page 84**
  - $\Theta(n+2)$
  - $\Theta(n + 100)$
157. Which symbol is used for Omega notation?
- (O)
  - ( $\Theta$ )
  - ( $\Omega$ )**
  - (@)
158. Selection sort is a\_\_\_\_\_sorting algorithm
- In-place page 54**
  - Not In-Place
  - Stable
  - in-partition
159. In Dynamic Programming based solution of knapsack problem, to compute entries of 'V', we will imply a(n)\_\_\_approach.
- Subjective
  - Inductive
  - Brute Force
  - Combination

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

160. We do not need to prove comparison-based sorting algorithms bymathematically. It always takes \_\_time.
- a. Big Oh  $n \log(n)$
  - b. Omega  $n \log(n)$
  - c. Omega  $n(n^2)$
  - d. Theta  $n \log(n)$



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

161. Merge sort is a/an \_\_\_\_\_ and \_\_\_\_\_ sorting algorithm
- Not in-place, not stable one
  - In-place, not stable one
  - In-place, stable one
  - Not in-place, stable one** page 54
162. Cubic function will \_\_\_\_\_ a quadratic function.
- Prove
  - be equal to
  - overtake** Page 25
  - find
163. Insertion sort is a \_\_\_\_\_ sorting algorithm
- Unstable
  - In-place** Page 54
  - Not In-Place
  - in-partition
164. To check whether a function grows faster or slower than the other function, we use some asymptotic notations, which is\_\_.
- Big-oh notation
  - Theta notation
  - Omega notation
  - All of the given
165. Asymptotic growth of  $8n^2 + 2n - 3$  is:
- $\Theta(n^2 + n)$
  - $\Theta(n^2)$**  page 14
  - $\Theta(8n^2)$
  - $\Theta(8n^2 + 2n)$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

166. In the analysis of algorithms, \_\_\_\_\_ plays an important role.
- a. text analysis
  - b. time
  - c. growth rate
  - d. money



## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

167. In inductive approach of knapsack problem, we consider 2 cases,  
\_\_\_Or \_\_\_\_.
- a. Median, Mode
  - b. Recursive, Iterative
  - c. **Leave object, Take object** page 93
  - d. Sequentially. Parallel
168. Random Access Machine (RAM) can execute \_\_\_\_\_ instructions
- a. only logical
  - b. parallel
  - c. only arithmetic
  - d. **logical and arithmetic**
169. Using \_\_\_\_\_ algorithm, efficiency is not given much importance
- a. Greedy
  - b. Merge sort
  - c. **Processing** as there is no algorithm by this name
  - d. Brute Force
170. Bubble sort takes theta \_\_\_\_\_ in the worst case
- a. **(n<sup>2</sup>)** page 39
  - b. (n)
  - c. log(n)
  - d. nlog(n)
171. If matrix A of dimension  $p \times q$  is multiply with matrix B of dimension  $q \times r$ , then dimension of resultant matrix is:
- a.  $p \times q$
  - b.  **$p \times r$**  page 84
  - c.  $q \times r$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- d.  $r \times p$
172. Dynamic Programming algorithms often use some kind of \_\_\_\_\_ to store the results of intermediate sub-problems
- a. table (Page 75)
  - b. variable
  - c. stack
  - d. loop



**VU Answer**  
your right path

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

173. \_\_\_\_\_ is in-place sorting algorithm.
- Bubble sort (Page 54)
  - Merge sort
  - Linear search
  - Binary Search
174. Which one of the following problems can be solved using dynamic problem?
- Bubble sort problem
  - Matrix chain multiplication problem page 85
  - Greedy search problem
  - Fractional knapsack problem
175. In chain matrix multiplication, solutions of the sub-problems are stored in a \_\_\_\_\_.
- Array
  - Table page 86
  - Tree
  - Link list
176. What is the average running time of a quick sort algorithm?
- $O(n^2)$
  - $O(n)$
  - $O(n \log n)$  (Page 49)
  - $O(\log n)$
177. Sorting Algorithms having  $O$ \_\_\_\_\_running time are considered to beslow ones.
- (n)

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

**b.  $(n^2)$  (Page 39)**

c.  $(n \log(n))$

d.  $(\log(n))$

178. While solving Selection problem, in Sieve technique we partition input data

a. In increasing order

b. In decreasing order

**c. According to Pivot**

d. Randomly

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

179. \_\_\_\_\_ is the process of avoiding unnecessary repetitions by writing down the results of recursive calls and looking them up again if we need them later.
- a. Loop
  - b. Memoization** page 74
  - c. Recursion
  - d. Function
180. In average-case time the probability of seeing input is denoted by \_\_.
- a.  $p\{l\}$
  - b.  $p[l]$
  - c.  $p\langle i \rangle$
  - d.  $p(i)$**  page 13
181. While applying the Sieve technique to selection sort, how to choose a pivotelement.
- a. Through mean
  - b. Linear
  - c. Randomly** page 35
  - d. Sequentially
182. Number of \_\_\_\_\_ of the pseudo code are counted to measure the runningtime.
- a. Inputs
  - b. Outputs
  - c. Steps** page 13
  - d. Pages
183. Developing a dynamic programming algorithm generally involves \_\_\_ separate steps.

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- a. One
  - b. **Two** page 75
  - c. Three
  - d. Four
184.  $8n^2+2n+3$  will exceed  $c28(n)$ , no matter how large we make\_.
- a. n
  - b.  $2n$
  - c. **c2** page 25
  - d. this quadratic equation

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

185. The running time of quick sort algorithm\_\_\_\_\_.
- Is impossible to compute
  - Has nothing to do with pivot selection
  - Is Random upon each execution
  - Greatly influenced by the selection of pivot page 49**
186. \_\_\_\_\_ involves breaking up the problem into sub problems whose solutions can be combined to solve the global problem.
- Complexity Theory
  - Dynamic programming solution
  - Divide and Conquer Strategy page 34**
  - Greedy Algorithms
187. In \_\_\_\_\_ we have to find rank of an element from given input.
- Merge sort algorithm
  - Selection problem page 34**
  - Brute force technique
  - Plane Sweep algorithm
188. How many steps are involved to design the dynamic programming strategy?
- 2
  - 3
  - 1
  - 4 page 92**
189. In Bucket sort, if there are duplicates then each bin can be replaced by a
- Stack
  - Linked list page 69**

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Hash table
  - d. Heap
190. In merge sort algorithm, we split the array \_\_\_ to find index q.
- a. from start
  - b. midway** page 28
  - c. from end
  - d. both from start or end

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

191. Find the maximum value of the items which can carry using knapsack. Knapsack weight capacity = 50.

| Item | Weight | Value |
|------|--------|-------|
| 1    | 10     | 70    |
| 2    | 20     | 20    |
| 3    | 30     | 80    |
| 4    | 70     | 200   |

- a. 280
- b. 100
- c. 90
- d. 200

192. In 2-d maxima problem a point p is said to be dominated by point q if

- a.  $p.x \leq q.x$
- b.  $p.x \leq q.x$  and  $p.y \leq q.y$  page 17
- c.  $p.y \leq q.y$
- d.  $p.x \geq q.x$  and  $p.y \geq q.y$

193. Sorting can be in \_\_\_\_\_.

- a. Increasing order only
- b. Decreasing order only
- c. Both increasing and decreasing order
- d. Random order

194. Recurrence can be described in terms of \_\_\_\_\_.

- a. Array
- b. Linear

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. **Tree** **page 31**  
d. Graph
195. The brute-force algorithm for 2D-Maxima runs in order  $O()$  time.
- a.  $n$   
b.  $n(\log n)$   
c.  **$n*n$**  **page 18**  
d.  $n^3$

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

196. In plane sweep approach of solving geometric problems, a \_\_\_\_\_ is swept across the plane.
- a. **Line** page 18
  - b. Plane
  - c. Cube
  - d. Box
197. Which of the following is calculated with Big Omega notation?
- a. Medium bounds
  - b. Upper bounds
  - c. **Lower bounds** Page - 25
  - d. Both upper and lower bounds
198. \_\_\_\_\_ is always based on divide and conquer strategy.
- a. Bubble sort
  - b. Selection sort
  - c. Pigeon sort
  - d. **Quick sort** page 46
199. If a matrix has three rows and two columns, then dimensions of matrix will be:
- a. **3x2**
  - b. 2x3
  - c. 3x3
  - d. 2x2
200. Asymptotic notations are used to describe \_\_\_\_\_ of an algorithm.
- a. Length
  - b. **running time** google
  - c. size

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- d. compile time
201. Catalan numbers are related the number of different \_\_ on 'n' nodes.
- a. Arrays
  - b. linked lists
  - c. binary trees
  - d. functions

page 85

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

202. Applying the sieve technique to selection problem, \_\_\_\_\_ element is picked from array.
- Output
  - Total
  - Input
  - Pivot** page 35
203. Dynamic Programming approach is usually useful in solving \_\_\_\_\_ problems.
- Normal
  - Optimization** google
  - Array
  - Loop
204. In recursive formulation of knapsack Problem:  
 $V [0, j] = \underline{\hspace{2cm}}$  for  $j \geq 0$
- 1
  - 0** page 93
  - 1
  - 2
205. \_\_\_\_\_ is a linear time sorting algorithm.
- Merge sort
  - Radix sort** page 71
  - Quick sort
  - Bubble sort
206. Quick sort is one of the \_\_\_\_\_ sorting algorithm.
- Fastest** page 19
  - Slowest

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- c. Major
  - d. Average
207. The time assumed for each basic operation to execute on RAM model of computation is\_.
- a. Infinite
  - b. Continuous
  - c. **Constant** page 10
  - d. Variable

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

208. In Sieve Technique, we know the item of interest.
- True
  - False page 34
209. While analyzing algorithms, \_\_\_\_\_ and \_\_\_\_\_ usually considered difficult to calculate.
- Finite, Infinite
  - Floor, ceiling google
  - Row, Column
  - Graph, Tree
210. While analysis of the brute-force maxima algorithm, an array sorted in the reverse order is the type of \_\_\_ case input.
- Best
  - Worst page 14
  - Somewhat bad
  - Average
211. \_\_\_\_\_ is not useful measure of central tendency of given input set especially when the distribution of values is highly skewed.
- Mean
  - Mode
  - Average
  - Median page 34
212. In asymptotical analysis of  $n(n-3)$  and  $4n*n$ , as  $n$  becomes large, the dominant (fastest growing) term is some constant times \_\_\_\_\_.

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- a.  $n+1$
- b.  $n-1$
- c.  $n$
- d.  $n*n$  **page 23**

213. In addition to passing in the array itself to Merge Sort algorithm, we will pass in other arguments which are indices.

- a. **Two** **P-38**
- b. Three
- c. Four
- d. Five

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

214. In 2d-maximal problem, a point is said to be if it is not dominated by any other point in that space.
- a. Member
  - b. Minimal
  - c. Maximal P-11**
  - d. Joint
215. Counting sort assumes that the numbers to be sorted are in the range \_\_\_\_\_.
- a. K to n where n is large
  - b. 1 to k where k is small (P-57)**
  - c. K to n where k is small
  - d. k to n where n is small
216. Insertion sort is an efficient algorithm for sorting a \_\_\_\_\_ number of elements
- a. Large P-39**
  - b. Small
  - c. Extra large
  - d. Medium
217. If the indices passed to merge sort algorithm are \_\_\_\_\_ then this means that there is only one element to sort.
- a. Small page 28**
  - b. Large
  - c. Equal
  - d. Not Equal
218. In Knapsack Problem, each item must be entirely accepted or rejected, is called \_\_\_\_\_ problem.

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

a. Fractional

**b. 0-1 P-92**

c. Linear

d. Optimal



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

219. If the time complexity of an algorithm is  $O(n)$ . then it is called \_  
\_\_timecomplexity.
- a. **Linear** **Wikipedia**
  - b. Constant
  - c. Average
  - d. Exponential
220. In the case of \_\_\_\_\_ analysis does not depend upon on the  
distribution of input.
- a. Merge sort
  - b. Quick sort** **P-50**
  - c. Insertion sort
  - d. Heap sort
221. We can use the \_\_\_\_\_ Property to devise a recursive  
formulation of the edit distance problem.
- a. Small substructure
  - b. Algorithmic
  - c. Real
  - d. Optimal substructure** **page 78**
222. The following sequence is called \_\_\_\_\_  
1,2,3,5,8,13,21,34,55,.....
- a. Optimize sequence
  - b. Fibonacci sequence** **page 73**
  - c. Optimal sequence
  - d. Overlapping sequence
223. Which one sorting algorithm is best suited to sort an array of 2  
million elements?

## SUBSCRIBE OUR CHANNEL

[https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWu  
WMIg3Q](https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q)

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- a. Bubble sort
- b. Insert sort
- c. Merge sort
- d. Quick sort

**e. Ridx Sort**      **page 71**



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

224. We can improve the performance of quick sort if we could be able to \_\_\_\_\_.
- a. Skip input elements somehow
  - b. Select two or more pivots** **page 34**
  - c. Skip any sub-array completely
  - d. Eliminate recursive calls
225. The problem with the brute-force algorithm is that it uses \_\_\_\_\_ in pruning out de
- a. Worst-case time
  - b. No intelligence** **page 18**
  - c. Outside looping
  - d. Artificial intelligence
226. In chain matrix multiplication, the order of the matrices \_\_\_\_\_.
- a. Can be changed
  - b. Can not be changed** **page 85**
  - c. is equal
  - d. is reverse
227. In quick sort algorithm, we choose pivot \_\_\_\_\_.
- a. Always the smallest element
  - b. Greater than 5
  - c. Randomly** **page 35**
  - d. Less than 5
228. In Heap Sort algorithm. Heapify procedure is \_\_\_\_\_ in nature.
- a. Recursive
  - b. Non-Recursive** **page 43**
  - c. Fast

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

d. Slow

229. When matrix A of  $5 \times 3$  is multiplied with matrix B of  $3 \times 4$  then the number of multiplications required will be \_\_\_\_.

a. 15

b. 12

c. 3

d. 60

60

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

230. An algorithm is said to be correct if for every \_\_\_\_\_ instance, it halts with the correct \_\_\_\_.
- a. **Input, Output** page 13
  - b. Design, Analysis
  - c. Value, Key
  - d. Key, Analysis
231. In chain matrix multiplication, table is filled \_\_\_\_\_ to find the multiplication of matrix.
- a. row wise
  - b. column wise
  - c. diagonally
  - d. **bottom-to-up** page 86
232. If we have an equation  $8n^2 + 7f \cdot n + 5f + 6$  then is large, \_\_\_\_\_ term will be much ~~xxxxxxx~~ the  $n$  term and will dominate the running time.
- a.  $f \cdot g(n)$
  - b.  $g(n) \cdot 2$
  - c.  **$n \cdot 2$**  page 23
  - d.  $f(n)$
233. For quick sort algorithm. Partitioning takes theta \_\_\_\_\_.
- a.  $(n)$
  - b.  $\log(n)$
  - c.  $n \log(n)$
  - d.  **$n^2 \log(n)$**
234. In Heap Sort algorithm, the maximum levels an element can move upward is

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- \_\_\_\_\_
- a. **Theta (log n)** page 43
  - b. Big-ch (log n)
  - c. Omega (log n)
  - d.  $O(1)$  i.e. Constant time
235. \_\_\_\_\_ programming is essentially recursion without repetition.
- a. Fast
  - b. **Dynamic** page 75
  - c. Array
  - d.  $n (\log n)$

**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

236. There are no hard formal rules to the syntax of the \_\_\_\_ code.
- a. Basic
  - b. Programmin
  - gc. **Pseudo**
  - d. Assembly
237. In Heap Sort algorithm, to remove the maximum element every time.
- a. We call Build-Heap procedure
  - b. Heap Sort algorithm terminates without resultc. **We call heapify procedure**
  - d. Nothing happens
238. Which process is used for avoiding unnecessary repetitions and lookingthem up again if we need them later.
- a. Greedy Approach
  - b. **Memoization** page 74
  - c. Divide and conquer
  - d. Recursion
239. The worst-case running time of Quick sort is \_\_\_\_\_ in order to sort anarray of n element.
- a.  **$O(n \log n)$**  page 49
  - b.  $O(n)$
  - c.  $O(n^2)$
  - d.  $O(\log n)$
240. Boolean operation is a \_\_\_\_\_ operation on an idealized RAM model ofcomputation.
- a. Starting

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>

**VISIT ANY MORE SOLUTIONS**

**[vuanswer.com](http://vuanswer.com)**

- b. **Basic** page 10
- c. Advance
- d. Normal



**SUBSCRIBE OUR CHANNEL**

**<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>**

**JOIN WHATSAPP GROUP**

**<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>**

## VISIT ANY MORE SOLUTIONS

[vuanswer.com](http://vuanswer.com)

241. In chain matrix multiplication, if there are  $n$  items, there are \_\_\_\_\_ ways in which outer most pair of parentheses can be placed.
- a.  $n^2$
  - b.  $2n$
  - c.  $n+1$
  - d.  $n-1$  **page 85**
242. The number of nodes in a complete binary tree of height  $h$  is:
- a.  $2 * (h+1) - 1$
  - b.  $2^{(h+1)} - 1$  **page 40**
  - c.  $2 * (h+1)$
  - d.  $((h+1)^2) - 1$

## SUBSCRIBE OUR CHANNEL

<https://www.youtube.com/channel/UCvFS8fAv4bvZsOmWuWMIg3Q>

## JOIN WHATSAPP GROUP

<https://chat.whatsapp.com/CbEPu7qfeNV8v0dZd5yT8J>



**CS502- Fundamentals of Algorithms**  
**Solved MCQS**  
**From Midterm Papers**

**May- 24 - 2013**

**MC100401285**

*Moaaz.pk@gmail.com*

*Mc100401285@vu.edu.pk*

**PSMD01**

**MIDTERM EXAMINATION**  
**Fall 2011**  
**CS502- Fundamentals of Algorithms**

**Question No: 1 ( Marks: 1 ) - Please choose one**

Due to left complete nature of binary tree, the heap can be stored in

- ▶ **Arrays (Page 40)**
- ▶ Structures
- ▶ Link Lis
- ▶ Stack

**Question No: 1 ( Marks: 1 ) - Please choose one**

What type of instructions Random Access Machine (RAM) can execute?

- ▶ Algebraic and logic
- ▶ Geometric and arithmetic
- ▶ **Arithmetic and logic (Page 10)**
- ▶ Parallel and recursive

**Question No: 1 ( Marks: 1 ) - Please choose one**

For Chain Matrix Multiplication we can not use divide and conquer approach because,

- ▶ **We do not know the optimum k (Page 86)**
- ▶ We use divide and conquer for sorting only
- ▶ We can easily perform it in linear time
- ▶ Size of data is not given

**Question No: 1 ( Marks: 1 ) - Please choose one**

What is the total time to heapify?

- ▶  **$O(\log n)$  (Page 43)**
- ▶  $O(n \log n)$
- ▶  $O(n^2 \log n)$
- ▶  $O(\log^2 n)$

**DOWNLOAD MIDTERM**

**PAST PAPERS BY WAQAR SIDDHU**

**More in PDF From**

**VU Answer**

**Get All Solutions.**

Al-Khwarizmi's work was written in a book titled \_\_\_\_\_.

Answer ( Please select your correct option )

al Kitab al-mukhtasar fi hisab al-jabrwa'l-muqabalah

Calculation by Completion

al Kitab

al-jabrwa'l-muqabalah



What is the asymptotic growth of  $\frac{4n^3 + 15n^2 + 11n}{6}$ ?

Answer ( Please select your correct option )

$\Theta\left(\frac{4n^3 + 15n^2 + 11n}{6}\right)$

$\Theta(4n^3 + 15n^2 + 11n)$

$\Theta(15n^2)$

$\Theta(n^3)$



The reason for introducing Sieve Technique algorithm is that it illustrates a very important special case of,

Answer ( Please select your correct option )

divide-and-conquer

✓

decrease and conquer

greedy nature

2-dimension Maxima

Strictly increasing arrays are \_\_\_\_\_ input for quick sort.

Answer ( Please select your correct option )

The worst-case



The best-case

Average case

None of the given

The running time of quick sort depends heavily on the selection of

Answer ( Please select your correct option )

Required Output

Input and output

Pivot

Input data only



Quick sort is based on divide and conquer paradigm. In the Combine step of divide and conquer process

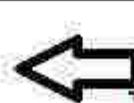
Answer ( Please select your correct option )

No work is needed to combine the sub-arrays, the array is already sorted

Work is needed to combine the sub-arrays, the array is not already sorted.

Merging the Arrays

Dividing the arrays



**Made By: Waqar Siddhu**

In in-place sorting algorithm is one that uses no \_\_\_\_\_ arrays for storage.

Answer ( Please select your correct option )

two dimensional

three dimensional

n dimensional

additional



**Made By: Waqar Siddhu**

Bubble sort and insertion sort can be implemented as stable algorithms but \_\_\_\_\_ can not be.

Answer ( Please select your correct option )

selection sort

correct answer solve  
by hadi

bubble sort

merge sort

stable sort

Catalan number is given by the formula

Answer ( Please select your correct option )

$C(n) = \frac{1}{n+1} \binom{2n}{n}$

correct answer solve  
by hadi

$C(n) = \frac{1}{n-1} \binom{2n}{n}$

$C(n) = \frac{1}{n+1} \binom{n}{2n}$

$C(n) = \frac{1}{n-1} \binom{n}{2n}$

We can find the product  $A \times B$  of matrices  $A$  and  $B$ , only if they are compatible which means,

Answer ( Please select your correct option )

No of Columns of  $A$  must be equal to No of Rows of  $B$

No of Columns of  $A$  must be equal to No of Columns of  $B$

No of Rows of  $A$  must be equal to No of Rows of  $B$

correct answer solve  
by hadi

Order of  $A$  must be equal to order of  $B$

Matrix - Chain - Order is \_\_\_\_\_ than the exponential time method of enumerating all possible parenthesizations and checking each one.

Answer ( Please select your correct option )

Much more efficient

Only fractional efficient

Worst

Too slow

Time complexity of chain matrix multiplication is  $\Theta(n^3)$  and space complexity is

Answer ( Please select your correct option )

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(n^3)$

$\Theta(n \log n)$

$\Theta(\log n)$

The Knapsack problem belongs to the domain of \_\_\_\_\_ problems.

Answer ( Please select your correct option )

Optimization

correct answer solve  
by hadi

NP Complete

Linear Solution

Sorting

The recurrence relation of Tower of Hanoi is given below

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(n-1)+1 & \text{if } n>1 \end{cases}$$

In order to move a tower of 4 rings from one peg to another, how many ring moves are required?

Answer ( Please select your correct option )

 15 7 12 32

Which type of instructions Random Access Machine (RAM) can execute? Choose best answer

Answer ( Please select your correct option )

Algebraic and logic

Geometric and arithmetic

Arithmetic and logic

Parallel and recursive

correct answer solve  
by hadi

Algorithm's essential elements are

Answer ( Please select your correct option )

Step wise solution

Stepwise solution and finite time

Step wise solution finite inputs

correct answer solve  
by hadi

Stepwise approach in which time and memory does not matter.

Which one is the correct conversion of the given algorithm to sigma notation?

```
MAXIMA(int n, Point P[1... n])
```

```
1 for i ← 1 to n
```

```
2 do maximal ← true
```

```
3 maximal ← false
```

Answer ( Please select your correct option )

$\sum_{i=1}^n 4i + 4$

$\sum_{i=1}^n 4i + 2$

$\sum_{i=1}^n 4i + 2i$

$\sum_{i=1}^n 2i + 2$

correct answer solve  
by hadi

Which of the following is calculated with **Big Theta** notation?

Answer ( Please select your correct option )

Lower bounds

Upper bounds

Both upper and lower bound

Medium bounds

correct answer solve  
by hadi

Suppose we have an algorithm that carries out  $N^3$  operations for an input of size  $N$ . Let us say that a computer takes 1 microsecond ( $1/1000000$  second) to carry out one operation. How long does the algorithm run for an input of size 3000?

Answer ( Please select your correct option )

90 seconds

9 seconds

correct answer solve  
by hadi

0.9 seconds

0.09 seconds

**Made By: Waqar Siddhu**

Which one is the best algorithm from the following with respect to running time?

Answer ( Please select your correct option )

$\Theta(n)$

$\Theta(n^2)$

$\Theta(n \log n)$

$\Theta(n^3)$

correct answer solve  
by hadi

Unlike programs, algorithms to be understood primarily by \_\_\_\_\_ and \_\_\_\_\_.

Answer ( Please select your correct option )

Machines, not people

Mathematical expressions, not algebraic expressions

Programmers, not machines

RAM, not programmer

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Recurrences are useful for analyzing

Answer ( Please select your correct option )

Recursive Algorithms

correct answer solve  
by hadi

Simple Algorithms

Parallel Algorithms

Parallel Algorithms & Recursive Algorithms

Divide-and-conquer involves breaking the problem into a small number of

Answer ( Please select your correct option )

pivot

Sub problems

correct answer solve  
by hadi

Selection

Sieve

In which order we can sort?

Answer ( Please select your correct option )

increasing order or decreasing order

correct answer solve  
by hadi

both at the same time

increasing order only

decreasing order only

Comparison based sorting algorithms can not run faster than

Answer ( Please select your correct option )

$\Omega(n \log n)$

correct answer solve  
by hadi

$(n \log n)$

$\Omega(n^2)$

$(n^2)$

In Quick sort, we don't have the control over the sizes of recursive calls

Answer ( Please select your correct option )

True

correct answer solve  
by hadi

False

Less information to decide

Either true or false

Who invented Quick sort procedure?

Answer ( Please select your correct option )

Hoare

correct answer solve  
by hadi

Sedgewick

Melroy

Coreman

Counting sort assumes that the numbers to be sorted are in the range 1 to  $k$ , where  $k$  is

Answer ( Please select your correct option )

Small

correct answer solve  
by hadi

Large

No restriction on  $k$

None of these

If there are  $\Theta(n^2)$  entries in edit distance matrix then the total running time is

Answer ( Please select your correct option )

$\Theta(1)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(n)$

$\Theta(n \log n)$

When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has \_\_\_\_\_ sub-problems.

Answer ( Please select your correct option )

Overlapping

Over costing

Optimized

None of these

A  $p \times q$  matrix  $A$  can be multiplied with a  $q \times r$  matrix  $B$ . The result will be a  $p \times r$  matrix  $C$ . In particular, for  $1 \leq i \leq p$  and  $1 \leq j \leq r$ ,

Answer ( Please select your correct option )

$C[i, j] = \sum_{k=1}^q A[i, k]B[k, j]$  ✓

$C[i, j] = \sum_{k=1}^q A[k, i]B[k, j]$

$C[i, j] = \sum_{k=1}^q A[k, i]B[j, k]$

None of these

We can find the product  $A \times B$  of matrices  $A$  and  $B$ , only if they are compatible which means,

Answer ( Please select your correct option )

No of Columns of  $A$  must be equal to No of Rows of  $B$

No of Columns of  $A$  must be equal to No of Columns of  $B$

No of Rows of  $A$  must be equal to No of Rows of  $B$

Order of  $A$  must be equal to order of  $B$

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Time complexity of chain matrix multiplication is  $\Theta(n^3)$  and space complexity is

Answer ( Please select your correct option )

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(n^3)$

$\Theta(n \log n)$

$\Theta(\log n)$

Computational model of sequential RAM is:

Answer ( Please select your correct option )

Parallel machines may be expensive to model and have more computational power than sequential RAM

Computational power of sequential RAM is same as that of parallel machines only time efficiency is achieved with parallel machines

Both first and second options are true for the statement

Less powerful computational wise than parallel machines

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

The worst case running time of the algorithm given below is.

```
MAXIMA(int n, Point P[1...n])
```

```
1 for i ← 1 to n
```

```
2 do maximal ← true
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{6}\right)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(2n \lg 6)$

```
1 for i ← 1 to n
2 do maximal ← true
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x < P[j].x) & (P[i].y < P[j].y)
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{6}\right)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(2n \lg 6)$

**Made By: Waqar Siddhu**

```
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
7 if maximal
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{6}\right)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(2n \lg 6)$

**Made By: Waqar Siddhu**

```
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y) return true
6 then maximal ← false break
7 if maximal
8 then output P[i].x, P[i].y return true
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{6}\right)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(2n \lg 6)$

**Made By: Waqar Siddhu**

In Sieve Technique, we know \_\_\_\_\_

Answer ( Please select your correct option )

Item of interest

Order of items

complexity of items

All items are of interest

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

In Random access machine, instructions are executed \_\_\_\_\_.

Answer ( Please select your correct option )

five at a time

infinite instructions at a time

one-by-one

parallel

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Which type of instructions Random Access Machine (RAM) can execute? Choose best answer

Answer ( Please select your correct option )

Algebraic and logic

Geometric and arithmetic

Arithmetic and logic

Parallel and recursive

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Which of the following is calculated with **Big O notation**?

Answer ( Please select your correct option )

Medium bounds

Lower bounds

Upper bounds

Both upper and lower bound

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Which of the following functions grows fastest as  $n$  gets larger?

Answer ( Please select your correct option )

$n^{10}2^n$

$n^83^n$

$n^55^n$

$n^32^{2n}$

Al-Khwarizmi's work was written in a book titled \_\_\_\_\_

Answer ( Please select your correct option )

al Kitab al-mukhatasar fi hisab al-jabrwa'l-muqabalah

correct answer solve  
by hadi

Calculation by Completion

al Kitab

al-jabrwa'l-muqabalah

Unlike programs, algorithms to be understood primarily by \_\_\_\_\_ and \_\_\_\_\_

Answer ( Please select your correct option )

Machines, not people

Mathematical expressions, not algebraic expressions

Programmers, not machines

RAM, not programmer

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Which formula is used for calculating worst case running time?

Answer ( Please select your correct option )

$T_{worst}(n) = \max_{|I|=n} T(I)$

correct answer solve  
by hadi

$T_{worst}(n) = \max_{|I|=1} T(I)$

$T_{worst}(n) = \max_{|I|=1} T(n)$

$T_{worst}(n) = \max_{|I|=n} T(n)$

Analysis of Selection algorithm ends up with,

page 37

Answer ( Please select your correct option )

$\Theta(n)$

correct answer solve  
by hadi

$\Theta(1 / 1 + n)$

$\Theta(n / 2)$

$\Theta((n / 2) + n)$

A (an) \_\_\_\_\_ is a left-complete binary tree that conforms to the *heap order*

p 40

Answer ( Please select your correct option )

heap

correct answer solve  
by hadi

binary tree

binary search tree

array

Principal operation for maintaining the heap property is called heapify, it is also called

p 43

Answer ( Please select your correct option )

sifting down

correct answer solve  
by hadi

sifting up

sifting left

sifting right

What is common between Bubble sort, Insertion sort, Selection sort, Quick sort, and Heap sort?

Answer ( Please select your correct option )

All are in-place algorithms

correct answer solve  
by hadi

All are stable algorithms

None of these

All are unstable algorithms

When primary memory storage is at a premium, a (an) \_\_\_\_\_ algorithm may be preferable.

:  
:(

Answer ( Please select your correct option )

In-place

Stable

External storage

None of these

correct answer solve  
by hadi

If there are  $\Theta(n^2)$  entries in edit distance matrix then the total running time is

Answer ( Please select your correct option )

$\Theta(1)$

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(n)$

$\Theta(n \log n)$

Catalan number is given by the formula

Answer ( Please select your correct option )

$C(n) = \frac{1}{n+1} \binom{2n}{n}$

correct answer solve  
by hadi

$C(n) = \frac{1}{n-1} \binom{2n}{n}$

$C(n) = \frac{1}{n+1} \binom{n}{2n}$

$C(n) = \frac{1}{n-1} \binom{n}{2n}$

For Chain Matrix Multiplication we can not use divide and conquer approach because,

Answer ( Please select your correct option )

We do not know the optimum k

correct answer solve  
by hadi

We use divide and conquer for sorting only

We can easily perform it in linear time

Size of data is not given

We can find the product  $A \times B$  of matrices A and B, only if they are compatible which means,

Answer ( Please select your correct option )

No of Columns of A must be equal to No of Rows of B

No of Columns of A must be equal to No of Columns of B

No of Rows of A must be equal to No of Rows of B

Order of A must be equal to order of B

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

Time complexity of chain matrix multiplication is  $\Theta(n^3)$  and space complexity is

Answer ( Please select your correct option )

$\Theta(n^2)$

correct answer solve  
by hadi

$\Theta(n^3)$

$\Theta(n \log n)$

$\Theta(\log n)$

The worst case running time of the algorithm given below is,

```
MAXIMA(int n, Point P[1...n])
```

```
1 for i ← 1 to n
2 do maximal ← true
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{n^6}\right)$

$\Theta(n^2)$

$\Theta(2n \lg 6)$

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

MAXIMA(int n, Point P[1...n])

```
1 for i ← 1 to n
2 do maximal ← true
3 for j ← 1 to n
4 do
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{n^6}\right)$

$\Theta(n^2)$

$\Theta(2n \lg 6)$

already submit,,,,correct  
answer solve by hadi

**Made By: Waqar Siddhu**

```
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
7 if maximal
```

Answer ( Please select your correct option )

  $\Theta(n^6)$   $\Theta\left(\frac{2n}{n^6}\right)$   $\Theta(n^2)$   $\Theta(2n \lg 6)$ 

correct answer solve  
by hadi...already  
submit

**Made By: Waqar Siddhu**

```
5 if (i ≠ j)&(P[i].x ≤ P[j].x)&(P[i].y ≤ P[j].y) return
6 then maximal ← false break
7 if maximal
8 then output P[i].x, P[i].y return
```

Answer ( Please select your correct option )

$\Theta(n^6)$

$\Theta\left(\frac{2n}{n^6}\right)$

$\Theta(n^2)$

$\Theta(2n \lg 6)$

correct answer solve  
by hadi...already  
submit

**Made By: Waqar Siddhu**

In Sieve Technique, we know \_\_\_\_\_

it applies to problems where we are interested in finding

a single item from a larger set of  $n$  items. We do not know which item is of interest,

repeat

Answer ( Please select your correct option )

Item of interest

Order of items

complexity of items

All items are of interest

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

In Random access machine, instructions are executed \_\_\_\_\_.

Answer ( Please select your correct option )

five at a time.

infinite instructions at a time.

one-by-one.

parallel.

correct answer solve  
by hadi

Which of the following is calculated with **Big Theta** notation?

Answer ( Please select your correct option )

Lower bounds

Upper bounds

Both upper and lower bound

Medium bounds

correct answer solve  
by hadi

Suppose we have an algorithm that carries out  $N^3$  operations for an input of size  $N$ . Let us say that a computer takes 1 microsecond ( $1/1000000$  second) to carry out one operation. How long does the algorithm run for an input of size 3000?

Answer ( Please select your correct option )

90 seconds

9 seconds

correct answer solve  
by hadi

0.9 seconds

0.09 seconds

Which one is the best algorithm from the following with respect to running time?

Answer ( Please select your correct option )

$\Theta(n)$

$\Theta(n^2)$

$\Theta(n \log n)$

$\Theta(n^3)$

correct answer solve  
by hadi

Which of the following is true?

Answer ( Please select your correct option )

The worst case time complexity of the quick sort is  $N \log N$

The best case time complexity of the merge sort is  $\log N$

The worst case time complexity of the selection sort is  $N^2$

The worst case time complexity of the merge sort is  $\log N$

correct answer solve  
by hadi

**Made By: Waqar Siddhu**

**Question No : 1 of 26** **Marks: 1 (Budgeted Time 1 Min)**

The word *Algorithm* comes from the name of the Muslim author \_\_\_\_\_

**Answer ( Please select your correct option )**

- Abu Ja'far Mohammad ibn Musa al-Khwarizmi or Al-Khwarizmi correct answer solve by hadi
- Khwarizm
- Kheva
- Uzbekistan

Start Time: 5:37 AM  
**60:00**  
Time Left



http://localhost/VUTES/client/Instructions.aspx

File Edit View Favorites Tools Help

Favorites Suggested Sites Web Slice Gallery

VU Examination System (CLIENT) VUTES 6.5 :::: Fall...

Page Safety Tools

Question No : 2 of 26

Marks: 1 (Budgeted Time 1 Min)

Al-Khwanizmi's work was written in a book titled \_\_\_\_\_

Answer ( Please select your correct option )

 al Kitab al-mukhatasar fi hisab al-jabrwa'l-muqabalah

correct answer solve  
by hadi

 Calculation by Completion al Kitab al-jabrwa'l-muqabalah

Start Time: 5:37 AM

60:00

Time Left



2



Question No : 3 of 26

Marks: 1 (Budgeted Time 1 Min)

\_\_\_\_\_ is a graphical representation of an algorithm

Answer ( Please select your correct option )

- $\Sigma$  notation
- $\Theta$  notation
- Flowchart
- Asymptotic notation

correct answer solve by hadi

Start Time: 5:37 AM

**60:00**

Time Left



3









**Question No : 4 of 26** **Marks: 1 (Budgeted Time 1 Min)**

What is the asymptotic growth of  $n^2 + \frac{4n + 15n + 11n}{16}$ ?

**Answer ( Please select your correct option )**

- $\Theta\left(\frac{4n + 15n + 11n}{16}\right)$
- $\Theta\left(n^2 + n + \frac{1}{16}\right)$
- $\Theta(n^3)$  correct answer solve by hadi
- $\Theta(n)$

Start Time: 5:37 AM  
**60:00**  
 Time Left



4



Question No : 5 of 26

Marks: 1 (Budgeted Time 1 Min)

Recurrences are useful for analyzing

Answer ( Please select your correct option )

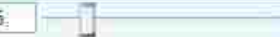
- Recursive Algorithms
- Simple Algorithms
- Parallel Algorithms
- Parallel Algorithms & Recursive Algorithms

correct answer solve  
by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left



5



Question No : 6 of 26 Marks: 1 (Budgeted Time 1 Min)

The sieve technique is a special case, where the number of sub-problems is just

Answer ( Please select your correct option )

- 5
- many
- 1
- few

correct answer solve by badi

Start Time: 5:37 AM

**60:00**

Time Left



6









Question No : 7 of 26

Marks: 1 (Budgeted Time 1 Min)

What is the total time to heapify?

Answer ( Please select your correct option )

- $O(\log n)$
- $O(n \log n)$
- $O(n^2 \log n)$
- $O(\log^2 n)$

correct answer solve  
by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

7

Navigation icons: info, back, forward, search

Question No : 8 of 26

Marks: 1 (Budgeted Time 1 Min)

Comparison based sorting algorithms can not run faster than

Answer ( Please select your correct option )

- $\Omega(n \log n)$
  - $\Theta(n \log n)$
  - $\Omega(n^2)$
  - $\Theta(n^2)$
- correct answer solve by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left



**Question No : 9 of 26** **Marks: 1 (Budgeted Time 1 Min)**

After partitioning array in Quick sort, pivot is placed in a position such that

**Answer ( Please select your correct option )**

- Values smaller than pivot are on left and larger than pivot are on right
- Values larger than pivot are on left and smaller than pivot are on right
- Pivot is the first element of array
- Pivot is the last element of array

correct answer solve by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

g [Progress Bar] [Info] [Previous] [Next] [Previous] [Next] [Print]

**Question No : 10 of 26** **Marks: 1 (Budgeted Time 1 Min)**

The running time of quick sort depends heavily on the selection of

**Answer ( Please select your correct option )**

- Required Output
  - Input and output
  - Pivot
  - Input data only
- correct answer solve  
by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

10

Navigation icons: Information, Previous, Home, Next, Previous, Next, Print

**Question No : 11 of 26** **Marks: 1 (Budgeted Time 1 Min)**

Memoization is a process

**Answer ( Please select your correct option )**

- To avoid unnecessary repetitions by writing down the results of recursive calls and looking them again if needed later
- To avoid repeated data by writing down the results of input array and looking them again if needed later.
- None of these
- All options are correct

correct answer solve  
by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

11

Navigation icons: info, back, forward, refresh

**Question No : 12 of 26** **Marks: 1 (Budgeted Time 1 Min)**

A problem exhibits \_\_\_\_\_ if an optimal solution to the problem contains within it optimal solution to sub-problems.

**Answer ( Please select your correct option )**

- Optimal structure correct answer solve by hadi
- Efficient structure
- Inefficient structure
- Unknown behavior

Start Time: 5:37 AM  
**60:00**  
Time Left



12



**Question No : 13 of 26** **Marks: 1 (Budgeted Time 1 Min)**

A  $p \times q$  matrix  $A$  can be multiplied with a  $q \times r$  matrix  $B$ . The result will be a  $p \times r$  matrix  $C$ . There are  $(p \cdot r)$  total entries in  $C$  and each takes \_\_\_\_\_ to compute.

**Answer ( Please select your correct option )**

- (a) correct answer solve by hadi
- (b)
- (c)
- (d)

Start Time: 5:37 AM  
**60:00**  
Time Left

13

Navigation icons: Information, Previous, Home, Next, Stop

Question No : 14 of 26

Marks: 1 (Budgeted Time 1 Min)

Catalan number is given by the formula

Answer ( Please select your correct option )

$$C(n) = \frac{1}{n+1} \binom{2n}{n}$$

correct answer solve  
by hadi

$$C(n) = \frac{1}{n-1} \binom{2n}{n}$$

$$C(n) = \frac{1}{n+1} \binom{n}{2n}$$

$$C(n) = \frac{1}{n-1} \binom{n}{2n}$$

Start Time: 5:37 AM

60:00

Time Left



14



**Question No : 15 of 26** **Marks: 1 (Budgeted Time 1 Min)**

For Chain Matrix Multiplication we can not use divide and conquer approach because,

**Answer ( Please select your correct option )**

- We do not know the optimum k correct answer solve by hadi
- We use divide and conquer for sorting only
- We can easily perform it in linear time
- Size of data is not given

Start Time: 5:37 AM  
**60:00**  
Time Left

15

Navigation icons: Information, Previous, Home, Next, Stop

**Question No : 16 of 26** **Marks: 1 (Budgeted Time 1 Min)**

Time complexity of chain matrix multiplication is  $\Theta(n^3)$  and space complexity is

**Answer ( Please select your correct option )**

- $\Theta(n^2)$  correct answer solve by hadi
- $\Theta(n^3)$
- $\Theta(n \log n)$
- $\Theta(\log n)$

Start Time: 5:37 AM  
**60:00**  
Time Left



Question No : 17 of 26

Marks: 1 (Budgeted Time 1 Min)

Computational model of sequential RAM is:

Answer ( Please select your correct option )

- Parallel machines may be expensive to model and have more computational power than sequential RAM
- Computational power of sequential RAM is same as that of parallel machines only time efficiency is achieved with parallel machines
- Both first and second options are true for the statement
- Less powerful computational wise than parallel machines

correct answer solve  
by hadi

Start Time: 5:37 AM

60:00

Time Left



17



**Question No : 18 of 26** **Marks: 1 (Budgeted Time 1 Min)**

What is the correct order of operations in merge sort?

**Answer ( Please select your correct option )**

- Conquer → Divide → Combine
- Divide → Conquer → Combine
- Conquer → Divide → Split
- Split → Conquer → Destroy

correct answer solve by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

18

Navigation icons: Information, Previous, Home, Next, Stop

Question No : 19 of 26

Marks: 1 (Budgeted Time 1 Min)

What is the average-case time for merge sort to sort an array of  $n$  elements?

Answer ( Please select your correct option )

- $O(\log n)$
- $O(n)$
- $O(n \log n)$
- $O(n^2)$

correct answer solve by hadi

Start Time: 5:37 AM  
**60:00**  
Time Left

19

Navigation icons: info, back, previous, next, forward, print

**Question No : 20 of 26** **Marks: 1 (Budgeted Time 1 Min)**

What is the solution to the recurrence  $T(n) = T(n/2) + n, T(1) = 1$ ?

**Answer ( Please select your correct option )**

- $O(\log n)$
- $O(n)$  correct answer solve by hadi
- $O(n \log n)$
- $O(n^2)$

Start Time: 5:37 AM  
**60:00**  
Time Left

20



We analyze an algorithm to check \_\_\_\_\_

Answer ( Please select your correct option )

- whether such an algorithm exists or not
- its running time
- it is a correct algorithm or not
- it will run on machine or not

correct answer solve  
by hadi

Start Time: 5:51 AM  
**60:00**  
Time Left

Navigation icons: Home, Back, Forward, Stop, Refresh

Question No : 2 of 26 Marks: 1 (Budgeted Time 1 Min)

What is the asymptotic growth of  $\frac{8n^2}{3} + \frac{n}{5}$ ?


Answer ( Please select your correct option )

- $\Theta\left(\frac{8n^2}{3}\right)$
- $\Theta\left(\frac{n}{5}\right)$
- $\Theta(n^2)$  correct answer solve by hadi
- $\Theta\left(\frac{8}{3}n\right)$

Start Time: 5:51 AM  
**60:00**  
Time Left



2



Question No : 3 of 26

Marks: 1 (Budgeted Time 1 Min)

What will be result of the following recurrence relation?

$$T(n) = \begin{cases} 4 & \text{if } n = 1 \\ T(n/5) + 3n^2 & \text{if } n \text{ is divisible by } 5 \end{cases}$$

Then T(5) is

Answer ( Please select your correct option )

- 25
- 75
- 79
- 70

correct answer solve by hadi

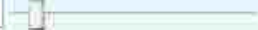
Start Time 5:51 AM

**60:00**

Time Left



3









**Question No : 4 of 26** **Marks: 1 (Budgeted Time 1 Min)**

Divide-and-conquer involves breaking the problem into a small number of

**Answer ( Please select your correct option )**

- Sieve
- pivot
- Sub problems
- Selection

correct answer solve  
by hadi

Start Time: 5:51 AM  
**60:00**  
Time Left

4

Navigation icons: Information, Previous, Home, Next, Stop

Question No : 5 of 26 Marks: 1 (Budgeted Time 1 Min)

The sieve technique is a special case, where the number of sub-problems is just

the sieve technique is a special case, where the number of subproblems is just 1.

Answer ( Please select your correct option )

- 5
- many
- 1 correct answer solve by hadi
- few

Start Time: 5:51 AM  
**60:00**  
Time Left



5 



When we call heapify then at each level the comparison performed takes time

Answer ( Please select your correct option )

- It will take  $\Theta(1)$
- Time will vary according to the nature of input data
- It can not be predicted
- It will take  $\Theta(\log n)$

correct answer solve  
by hadi

Start Time: 5:51 AM  
**60:00**  
Time Left

6

Navigation icons: info, back, forward, search, refresh

**Question No : 7 of 26** **Marks: 1 (Budgeted Time 1 Min)**

In Quick sort, we don't have the control over the sizes of recursive calls

**Answer ( Please select your correct option )**

- True correct answer solve by hadi
- False
- Less information to decide
- Either true or false

Start Time: 5:51 AM  
**60:00**  
Time Left



7



**Question No : 8 of 26** **Marks: 1 (Budgeted Time 1 Min)**

The running time of quick sort depends heavily on the selection of

**Answer ( Please select your correct option )**

- Required Output
- Input and output
- Pivot
- Input data only

correct answer solve by hadi

Start Time: 5:51 AM  
**60:00**  
Time Left

8

Navigation icons: Information, Previous, Home, Next, Previous, Next, Print

Question No.: 9 of 26

Marks: 1 (Budgeted Time 1 Min)

Any binary tree of height  $T(n)$  has at most  $2^{T(n)}$  leaves. Thus a comparison-based sorting algorithm can distinguish between at most \_\_\_\_\_ different final outcomes.

Answer ( Please select your correct option )

- $2^{T(n)}$  correct answer solve by hadi
- $2^{\log(n)}$
- $2^{n-1}$
- $2^n$

Start Time: 5:51 AM  
**60:00**  
Time Left



9



**Question No : 10 of 25** **Marks: 1 (Budgeted Time 1 Min)**

If there are  $\Theta(n^2)$  entries in edit distance matrix then each entry  $E(i, j)$  takes \_\_\_\_\_ time to compute:

**Answer ( Please select your correct option )**

- $\Theta(1)$  correct answer solve by hadi
- $\Theta(n^2)$
- $\Theta(n)$
- $\Theta(n \log n)$

Start Time 5:51 AM  
**60:00**  
Time Left 

10 

Question No: 11 of 25 Marks: 1 (Budgeted Time 1 Min)

If there are  $\Theta(n^2)$  entries in edit distance matrix then the total running time is

Answer ( Please select your correct option )

- $\Theta(1)$
- $\Theta(n^2)$  correct
- $\Theta(n)$
- $\Theta(n \log n)$

Start Time: 5:51 AM  
**60:00**  
Time Left



11




**Question No : 12 of 25** **Marks: 1 (Budgeted Time 1 Min)**

When a recursive algorithm revisits the same problem over and over again, we say that the optimization problem has \_\_\_\_\_ sub-problems:

**Answer ( Please select your correct option )**

- Overlapping ✓
- Over costing
- Optimized
- None of these

Start Time 5:51 AM  
**60:00**  
Time Left 

12 

**Question No : 13 of 25** **Marks: 1 (Budgeted Time 1 Min)**

A problem exhibits optimal structure if an optimal solution to the problem contains within it optimal solution to \_\_\_\_\_

**Answer ( Please select your correct option )**

- Sub-problems ✓
- All similar problems
- Efficient sorting algorithm
- Any unknown problem

Start Time 5:51 AM  
**60:00**  
Time Left 

13 

The Knapsack problem belongs to the domain of \_\_\_\_\_ problems.

Answer ( Please select your correct option )

- Optimization correct answer solve by hadi
- NP Complete
- Linear Solution
- Sorting

Start Time: 5:51 AM  
**60:00**  
Time Left

14

Navigation icons: info, back, forward, search, print

**Question No : 15 of 25** **Marks: 1 (Budgeted Time 1 Min)**

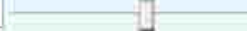
Computational model of sequential RAM is:

**Answer ( Please select your correct option )**

- Parallel machines may be expensive to model and have more computational power than sequential RAM
- Computational power of sequential RAM is same as that of parallel machines only time efficiency is achieved with parallel machines
- Both first and second options are true for the statement
- Less powerful computational wise than parallel machines

correct answer solve  
by hadi

Start Time 5:51 AM  
**60:00**  
Time Left 

15 

**Question No : 16 of 25** **Marks: 1 (Budgeted Time 1 Min)**

Merge sort requires

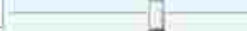
**Answer ( Please select your correct option )**

- Extra time than other sorting algorithms.
- Extra storage correct answer solve by hadi
- No need of extra storage
- Sometimes requires extra storage

Start Time: 5:51 AM  
**60:00**  
Time Left



16



Question No : 17 of 25

Marks: 1 (Budgeted Time 1 Min)

The worst case running time of the algorithm given below is,

```
MAXIMA(int n, Point P[1...n])
1 for i ← 1 to n
2 do maximal ← true
```

Answer ( Please select your correct option )

- $\Theta(n^6)$
- $\Theta\left(\frac{2n}{n^6}\right)$
- $\Theta(n^2)$
- $\Theta(2n \lg 6)$

correct answer solve  
by hadi

Start Time 5:51 AM  
**60:00**  
Time Left

17

Navigation icons: info, back, forward, search, refresh

Question No : 17 of 26 Marks: 1 (Budgeted Time 1 Min)

```
1
2 do maximal ← true
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
```

Answer ( Please select your correct option )

- $\Theta(n^6)$
- $\Theta\left(\frac{3n}{n^6}\right)$
- $\Theta(n^3)$
- $\Theta(2n \lg 6)$

correct answer solve  
by hadi

Start Time 5:51 AM 17 i [Navigation Icons]

**60:00** Time Left

Question No : 17 of 26 Marks: 1 (Budgeted Time 1 Min)

```
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
7 if maximal
```

Answer ( Please select your correct option )

- $\Theta(n^6)$
- $\Theta\left(\frac{3n}{n^6}\right)$
- $\Theta(n^3)$
- $\Theta(2n \lg 6)$

same back,,,correct answer solve by hadi

Start Time 5:51 AM  
**60:00**  
Time Left

17

Navigation icons: info, back, forward, search, etc.

Question No : 17 of 26 Marks: 1 (Budgeted Time 1 Min)

```

5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
7 if maximal
8 then output P[i].x, P[i].y

```

Answer ( Please select your correct option )

- $\Theta(n^6)$
- $\Theta\left(\frac{3n}{n^6}\right)$
- $\Theta(n^2)$  same back,,,correct  
answer solve by  
hadi
- $\Theta(2n \lg 6)$

Start Time 5:51 AM  
60:00  
 Time Left

17

Question No : 18 of 25 Marks: 1 (Budgeted Time 1 Min)

Which one is the correct conversion of the given algorithm to sigma notation?

```

MAXIMA(int n, Point P[1... n])
1 for i ← 1 to n
2 do maximal ← true

```

Answer ( Please select your correct option )

- $\sum_{i=1}^n 4i + 4$
- $\sum_{i=1}^n 4i + 2$  correct answer solve by hadi
- $\sum_{i=1}^n 4i + 2i$
- $\sum_{i=1}^n 2i + 2$

Start Time: 5:51 AM  
60:00  
 Time Left

18

Question No : 18 of 26 Marks: 1 (Budgeted Time 1 Min)

```
1 for i ← 1 to n
2 do maximal ← true
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
```

Answer ( Please select your correct option )

- $\sum_{i=1}^n 4n + 4$
- $\sum_{i=1}^n 4n + 2$  correct answer solve by hadi
- $\sum_{i=1}^n 4n + 2n$
- $\sum_{i=1}^n 2n + 2$

Start Time 5:51 AM  
**60:00**  
Time Left

18



Question No : 18 of 26 Marks: 1 (Budgeted Time 1 Min)

```
3 for j ← 1 to n
4 do
5 if (i ≠ j) & (P[i].x ≤ P[j].x) & (P[i].y ≤ P[j].y)
6 then maximal ← false break
7 if maximal
```

Answer ( Please select your correct option )

- $\sum_{i=1}^n 4n + 4$
- $\sum_{i=1}^n 4n + 2$  correct answer solve by hadi
- $\sum_{i=1}^n 4n + 2n$
- $\sum_{i=1}^n 2n + 2$

Start Time 5:51 AM  
**60:00**  
Time Left

18



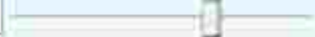
Question No : 18 of 26 Marks: 1 (Budgeted Time 1 Min)

```
5 if (i ≠ j)&(P[i].x ≤ P[j].x)&(P[i].y ≤ P[j].y) Correct
6 then maximal ← false break
7 if maximal
8 then output P[i].x, P[i].y Correct
```

Answer ( Please select your correct option )

- $\sum_{i=1}^n 4n + 4$
- $\sum_{i=1}^n 4n + 2$  correct answer solve by hadi
- $\sum_{i=1}^n 4n + 2n$
- $\sum_{i=1}^n 2n + 2$

Start Time 5:51 AM  
**60:00**  
Time Left 

18 

**Question No : 19 of 25** **Marks: 1 (Budgeted Time 1 Min)**

When this relation will be true?  $f(n) = O(f(n)^2)$

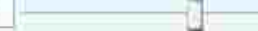
**Answer ( Please select your correct option )**

- Always True ✓
- Never True
- Some times True only
- True for small values of n

Start Time: 5:51 AM  
**60:00**  
Time Left



19



**Question No : 20 of 25** **Marks: 1 (Budgeted Time 1 Min)**

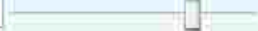
Which of the following functions grows fastest as  $n$  grows larger?

**Answer ( Please select your correct option )**

- $n^2$  correct answer solve by hadi
- $\log n$
- $n \log n$
- $2^n$

Start Time: 5:51 AM  
**59:00**  
Time Left



20 

**DOWNLOAD MIDTERM**

**PAST PAPERS BY WAQAR SIDDHU**

**More in PDF From**

**VU Answer**

**Get All Solutions.**



How edit distance is used for speech reorganization?

Answer ( [Please click here to Add Answer](#) )



What will be the sorting techniques by analyzing the followings?

- (i) Whenever two consecutive items are found in an array that is out of order, swap them.
- (ii) Assume that  $A[1 \dots i - 1]$  have already been sorted. Add  $A[i]$  into its proper position in this sub array. Create this position by shifting all larger elements to the right.
- (iii) Assume that  $A[1 \dots i - 1]$  contain the  $i - 1$  smallest elements in sorted order. Find the smallest element in  $A[i \dots n]$  Swap it with  $A[i]$ .

Answer ( [Please click here to Add Answer](#) )



True or False: A sequence of values in a row of the dynamic programming table for an instance of the knapsack problem is always non-decreasing. Give a brief description.

Answer ( [Please click here to Add Answer](#) )



Draw the final **Max-Heap** structure for the following array.

50, 31, 45, 30, 2, 7, 40, 12, 28, 1

You can show the final result (tree) only.

Answer ( [Please click here to Add Answer](#) )



Describe three solution paths for edit distance of these two given words,

1. MATHS
2. ARTS

Answer ( [Please click here to Add Answer](#) )



What is the essential constraint for the Counting Sort?

Answer ( [Please click here to Add Answer](#) )



How we proceed with  $m$  entries in cost table for chain matrix multiplication problem?

Answer ( [Please click here to Add Answer](#) )



Solve it,

$$T(n) = \frac{1}{2} \sum_{q=1}^2 (T(q-1) + T(2-q) + 2)$$

Answer ( [Please click here to Add Answer](#) )

A rich text editor toolbar with various icons for text formatting and editing. The toolbar includes icons for undo, redo, bold, italic, underline, strikethrough, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, and a help icon. The font size is set to 12 and the zoom level is 100%.

True or False: A sequence of values in a row of the dynamic programming table for an instance of the knapsack problem is always non-decreasing. **Give a brief description.**

Answer ( [Please click here to Add Answer](#) )



Whether each of the following sorting algorithms is stable and in-place or not?

| Sorting Algorithm | Stable | In-Place |
|-------------------|--------|----------|
| Merge Sort        | Yes/No | Yes/No   |
| Heapsort          | Yes/No | Yes/No   |
| Quicksort         | Yes/No | Yes/No   |
| Counting Sort     | Yes/No | Yes/No   |

Answer ( [Please click here to Add Answer](#) )



Normal Arial 12 B I U

|               |        |        |
|---------------|--------|--------|
| Merge Sort    | Yes/No | Yes/No |
| Heapsort      | Yes/No | Yes/No |
| Quicksort     | Yes/No | Yes/No |
| Counting Sort | Yes/No | Yes/No |
| Bubble Sort   | Yes/No | Yes/No |

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. It includes a font color selector, background color selector, bold (B), italic (I), underline (U), strikethrough, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, and a zoom percentage dropdown set to 100%.

Write down the steps of dynamic programming strategy.

Answer ( [Please click here to Add Answer](#) )



What is the necessary assumption for average case analysis of quick sort?

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. From left to right, the icons include: a folder icon, a document icon, a printer icon, a magnifying glass icon, a percentage icon, a list icon, a link icon, a text color icon, a background color icon, a bold icon, an italic icon, a strikethrough icon, a text color icon, a background color icon, a question mark icon, and a dropdown menu showing '100%'. Below the icons are dropdown menus for 'Normal' and 'Arial', and a dropdown menu for '12'. To the right of these are icons for bold, italic, strikethrough, bulleted list, numbered list, and indent.

If we solve the knap-sap algorithm with brute force then how much running time will be required?

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. From left to right, the icons include: a document icon, a folder icon, a magnifying glass icon, a percentage icon, a list icon, a link icon, a text color icon, a background color icon, a bold icon, an italic icon, a strikethrough icon, a text color icon, a background color icon, a question mark icon, and a dropdown menu showing '100%'. Below the icons, there are dropdown menus for 'Normal', 'Arial', and '12', followed by icons for bold, italic, strikethrough, and text alignment (left, center, right, justified).

In comparison based sorting algorithms, there are  $n!$  Permutations to arrange the elements, and lower bound achieved is  $\Omega(n \log n)$ . To prove this, which factors of binary tree enable us to take the "log" for  $n!$  comparisons?

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. From left to right, the icons include: a document icon, a folder icon, a magnifying glass icon, a percentage icon, a list icon, a link icon, a text color icon, a background color icon, a bold icon, an italic icon, a strikethrough icon, a text color icon, a background color icon, a question mark icon, and a dropdown menu showing '100%'. Below the icons, there are dropdown menus for 'Normal' and 'Arial', and a dropdown menu for '12'. To the right of these are icons for bold, italic, strikethrough, bulleted list, numbered list, and indent.

What are Catalan numbers? Give the formula to find Catalan numbers.

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. From left to right, the icons include: a document icon, a folder icon, a magnifying glass icon, a percentage icon, a list icon, a link icon, a text color icon, a background color icon, a bold icon, an italic icon, a strikethrough icon, a bulleted list icon, a numbered list icon, a decrease indent icon, an increase indent icon, a link icon, a unlink icon, a help icon, and a zoom dropdown menu set to 100%.

Consider the following recursive `search` function which returns the index of the array element containing `key`, if such an element exists.

```
int search(int* array, int left, int right, int key)
{
 int mid = (left+right)/2;
```

Answer ( [Please click here to Add Answer](#) )



A rich text editor toolbar with various icons for text formatting and editing. From left to right, the icons include: a folder icon, a document icon, a printer icon, a magnifying glass icon, a percentage icon, a list icon, a link icon, a text color icon, a background color icon, a bold icon, an italic icon, a strikethrough icon, a text color dropdown menu, a background color dropdown menu, a font size dropdown menu, a bold icon, an italic icon, a strikethrough icon, a bulleted list icon, an ordered list icon, a link icon, and a text color icon. The font size dropdown menu is set to 12, and the percentage icon is set to 100%.

```
int mid = (left+right)/2;
if(left == right)
 return left;
else if(array[mid] <= key)
 return search(array, mid+1, right, key);
```

Answer ( [Please click here to Add Answer](#) )



```
else if(array[mid] <= key)
 return search(array, mid+1, right, key);
else
 return search(array, left, mid, key);
}
```

Answer ( [Please click here to Add Answer](#) )



```
return search(array, left, mid, key);
}
```

Given the following declaration

```
int list[4] = { 5, 6, 7, 8 };
```

does `search(list, 0, 4, 8)` correctly return the index of the `list` element containing 8? Show your work.

Answer ( Please [click here](#) to Add Answer )



A rich text editor toolbar with various icons for text formatting and editing. It includes a dropdown menu for font style (currently set to 'Normal'), a dropdown for font color (currently set to 'Arial'), a dropdown for font size (currently set to '12'), and buttons for bold, italic, underline, strikethrough, bulleted list, numbered list, and link. A zoom level dropdown is also present, currently set to '100%'.



Question No : 21 of 26 Marks: 2 (Budgeted Time 4 Min)

What is the idea behind in counting sort algorithm, to sort the elements without comparisons in linear time?


Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:37 AM  
**60:00**  
Time Left



21



Question No : 22 of 26 Marks: 2 (Budgeted Time 4 Min)


Which one is the better approach to cop with chain matrix multiplication?

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:37 AM  
**60:00**  
Time Left



22. 

**Question No : 23 of 26** **Marks: 3 (Budgeted Time 6 Min)**

Consider sorting three numbers a1, a2 and a3 with comparison-based sorting algorithms then write the possible combinations?

**Answer ( Please click here to Add Answer )**

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:37 AM  
**60:00**  
Time Left

23

Navigation icons: Information, Previous, Next, Previous, Next, and Print.

Question No : 24 of 26 Marks: 3 (Budgeted Time 6 Min)

What are Catalan numbers? Give the formula to find Catalan numbers.

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:37 AM  
60:00  
Time Left

24 [Progress bar] [Info icon] [Previous question] [Next question]

Question No : 25 of 26 Marks: 5 (Budgeted Time 10 Min)

The following summation form represents quick sort, so what will be the outcome after solving this recurrence in terms of asymptotic notation?

$$\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$$

Answer ( [Please click here to Add Answer](#) )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The editor area is currently empty.

Start Time: 5:37 AM  
**60:00**  
Time Left



25       

Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

What is the time complexity of the following algorithm?

```
KNAPSACK(n, W)
1 for w = 0, W
2 do V[0, w] ← 0
3 for i = 0, n
4 do V[i, 0] ← 0
5 for w = 0, W
```

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The editor area is currently blank.

Start Time: 5:37 AM  
**60:00**  
Time Left

26

Navigation icons: Information, Previous, Next, Previous Page, Next Page.

Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

```
1 do $V[0, w] \leftarrow v$
2 for $i = 0, n$
3 do $V[i, 0] \leftarrow 0$
4 for $w = 0, W$
5 do if $(w_i \leq w \ \& \ v_i + V[i-1, w-w_i] > V[i-1, w])$
6 then $V[i, w] \leftarrow v_i + V[i-1, w-w_i]$
7 else $V[i, w] \leftarrow V[i-1, w]$
```

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:37 AM  
**60:00**  
Time Left

26 [Progress bar] [Info] [Previous] [Next] [Flag]

Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

```
4 do $V[i, w] \leftarrow v$
5 for $w = 0, W$
6 do if $(w_i \leq w \ \& \ v_i + V[i-1, w-w_i] > V[i-1, w])$
7 then $V[i, w] \leftarrow v_i + V[i-1, w-w_i]$
8 else $V[i, w] \leftarrow V[i-1, w]$
```

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The editor area is currently empty.

Start Time: 5:37 AM  
**60:00**  
Time Left

26

Navigation icons: Information, Previous, Next, Home, and Print.

Question No : 21 of 26

Marks: 2 (Budgeted Time 4 Min)

What is the idea behind in counting sort algorithm, to sort the elements without comparisons in linear time?

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:51 AM  
**59:00**  
Time Left

21 | [Info] [Previous] [Next] [Flag]

Question No : 22 of 26 Marks: 2 (Budgeted Time 4 Min)

How edit distance is used for speech reorganization?

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:51 AM  
59:00 Time Left

22 [Progress bar] [Info] [Previous] [Next] [Home]

Question No : 23 of 26 Marks: 3 (Budgeted Time 6 Min)

How to construct an Optimal Solution for 0/1 knapsack problem?

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:51 AM  
59:00 Time Left

23 [Progress bar] [Info icon] [Previous question] [Next question]

Question No : 24 of 26 Marks: 3 (Budgeted Time 6 Min)

In comparison based sorting algorithms, there are  $n!$  Permutations to arrange the elements, and lower bound achieved is  $\Omega(n \log n)$ . To prove this, which factors of binary tree enable us to take the "log" for  $n!$  comparisons?

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The text area below is empty.

Start Time: 5:51 AM  
**59:00**  
Time Left

24 [Progress bar] [Info] [Previous] [Next] [Home]

**Question No : 25 of 26** **Marks: 5 (Budgeted Time 10 Min)**

The table below shows the contents of an unsorted array. Perform radix sort on this array, show the contents of the array after the first and second pass of the sort.

| Unsorted Data | After Pass 1 | After Pass 2 |
|---------------|--------------|--------------|
| 342           |              |              |
| 103           |              |              |
| ...           |              |              |

**Answer ( Please click here to Add Answer )**

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level. The editor area is currently empty.

Start Time: 5:51 AM  
**59:00**  
Time Left

Progress bar showing question 25 of 26. Navigation buttons: Information, Previous, Next, Previous Page, Next Page.

Question No : 25 of 26 Marks: 5 (Budgeted Time 10 Min)

|     |  |  |  |
|-----|--|--|--|
| 342 |  |  |  |
| 103 |  |  |  |
| 211 |  |  |  |
| 440 |  |  |  |
| 332 |  |  |  |
| 441 |  |  |  |
| ... |  |  |  |

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Rich text editor menu bar with options: Normal, Arial, 12, B, I, U, and various list and alignment icons.

Large empty text area for entering the answer.

Start Time: 5:51 AM

**59:00**  
Time Left



25



Question No : 25 of 26 Marks: 5 (Budgeted Time 10 Min)

|     |  |  |
|-----|--|--|
| 200 |  |  |
| 210 |  |  |
| 402 |  |  |
| 312 |  |  |
| 124 |  |  |

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Rich text editor toolbar with dropdowns for font face (Normal, Arial) and font size (12), and icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Start Time: 5:51 AM

**59:00**  
Time Left



25



Question No : 25 of 26 Marks: 5 (Budgeted Time 10 Min)

|     |  |  |
|-----|--|--|
| 332 |  |  |
| 441 |  |  |
| 200 |  |  |
| 210 |  |  |
| 402 |  |  |
| 312 |  |  |

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Rich text editor menu bar with options: Normal, Arial, 12, B, I, U, and list/indent icons.

Large empty text area for providing the answer.

Start Time: 5:51 AM

**59:00**  
Time Left



25

Navigation icons: Information (i), Previous question, Previous question, Next question, Next question, and Print.

Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

Complete the following instance of the optimal matrix multiplication ordering problem,  
A1 A2 A3 A4 A5 A6  
(6x5) (5x2) (2x8) (8x4) (4x3) (3x9)

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Start Time: 5:51 AM  
**59:00**  
Time Left

26 i [Navigation icons]

Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 2 | 2 | ? |
|   | 0 | 2 | 2 | 2 | 2 |
|   |   | 0 | 3 | 4 | 5 |
|   |   |   | 0 | 4 | 5 |
|   |   |   |   | 0 | 5 |
|   |   |   |   |   | 0 |

Answer ( Please click here to Add Answer )

Rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Indent, Outdent, Undo, Redo, and a 100% zoom level.

Start Time: 5:51 AM  
**59:00**  
Time Left

26 ▶ i ▶ ▶ ▶ ▶ ▶

**Question No : 26 of 26** **Marks: 5 (Budgeted Time 10 Min)**

|   |    |     |     |     |     |
|---|----|-----|-----|-----|-----|
| 0 | 60 | 156 | 172 | 184 | ?   |
|   | 0  | 80  | 104 | 118 | 232 |
|   |    | 0   | 64  | 88  | 142 |
|   |    |     | 0   | 96  | 312 |
|   |    |     |     | 0   | 108 |
|   |    |     |     |     | 0   |

**Answer ( Please click here to Add Answer )**

100%

Normal Arial 12 B I U

Start Time: 5:51 AM

**59:00**  
Time Left



26



