CLASS TEST S.No.:01SKCS_ABCDEFG_230624

MADE EASY
India's Best Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Kolkata

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

ALGORITHMS

COMPUTER SCIENCE & IT

Date of Test : 23/06/2024

ANSWER KEY

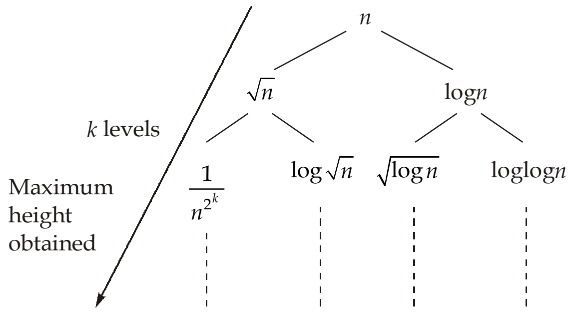
1.	(b)	7.	(a)	13.	(d)	19.	(c)	25.	(c)
2.	(a)	8.	(b)	14.	(b)	20.	(b)	26.	(b)
3.	(c)	9.	(d)	15.	(c)	21.	(d)	27.	(b)
4.	(c)	10.	(b)	16.	(d)	22.	(b)	28.	(c)
5.	(b)	11.	(b)	17.	(d)	23.	(b)	29.	(a)
6.	(a)	12.	(a)	18.	(b)	24.	(d)	30.	(d)

DETAILED EXPLANATIONS

1. **(b)**

Binary search → Divide and conquer
Quick sort → Divide and conquer
Insertion sort → Comparison based
0/1 Knapsack problem → Dynamic programming

2. **(a)**



So,
$$n^{\frac{1}{2^k}} = 2$$
$$\frac{1}{2^k}\log n = 1$$
$$\Rightarrow \quad \log n = 2^k$$
$$k = \log\log n = \theta(\log\log n)$$

3. **(c)**

S_1: Quick sort only has divide step and conquer step.
S_2: True: Since merge sort is an out place algorithm.
S_3: True: Number of comparisons in partition algorithm do not change for best case or worst case.
S_4: True

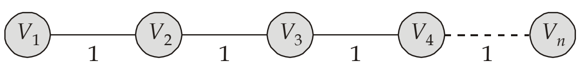
4. **(c)**

S_1: False
S_2: True; as the memory is not contiguous searching an element will take O(n) time even if we apply binary search [because of traversing].
S_3: True: First sort the array using merge sort then apply binary search.
S_4: True: Apply linear search.

5. **(b)**

Prim's algorithm will pick up the edge with least weight for a particular node such that it does not form a cycle.

∴ MST will be



Total n – 1 edges will be required.

So, weight = $5(n - 1)$

6. **(a)**

- Linear probing suffers from primary clustering.
- Inserting a key require unsuccessful search by placing the key into the first empty slot found.

Thus the expected number of probes is at most $\frac{1}{1-\alpha}$.

7. **(a)**

$$9000 + 112500 = 121500$$

E $\rightarrow$ has 500 columns so it should be multiplied only once, in the last.

A B C D

- (A(B C D))
  - (A(B (C D))) = 250 × 15 + 10 × 15 × 20 + 15 × 20 × 15 = 11250
  - (A((B C) D)) = 20 × 10 × 25 + 20 × 25 × 15 + 15 × 20 × 15 = 17000
- (A B) (C D) = 15 × 20 × 10 + 10 × 25 × 15 + 15 × 10 × 15 = 9000
- ((A B C) D)
  - (((A) (B C)) D) = 20 × 10 × 25 + 25 × 20 × 15 + 15 × 25 × 15 = 18125
  - (((A B) C) D) = 15 × 20 × 10 + 15 × 10 × 25 + 15 × 25 × 15 = 12375

8. **(b)**

String 1 $\rightarrow$ *a a b b c p q r q p*

String 2 $\rightarrow$ *s a b a b t q c p t s p q*

		a	*a*	*b*	*b*	*c*	*p*	*q*	*q*	*p*
	0	0	0	0	0	0	0	0	0	0
a	0	1	1	1	1	1	1	1	1	1
b	0	1	1	2	2	2	2	2	2	2
a	0	1	2	2	2	2	2	2	2	2
b	0	1	2	3	3	3	3	3	3	3
q	0	1	2	3	3	3	3	4	4	4
c	0	1	2	3	3	4	4	4	4	4
p	0	1	2	3	3	4	5	5	5	5
p	0	1	2	3	3	4	5	5	5	6
q	*o*	1	2	3	3	4	5	6	6	6

Maximum length = 6

$$\begin{bmatrix} a\ a\ b\ c\ p\ q \\ a\ a\ b\ c\ p\ p \\ a\ b\ b\ c\ p\ q \\ a\ b\ b\ c\ q\ p \end{bmatrix} \rightarrow \text{possible 6 – length strings.}$$

9. **(d)**

$$e_1 + e_2 + e_3 + e_4 = 8$$

$$^{8+4-1}C_{4-1}$$

$e_1, e_2, e_3, e_4 \geq 0$

$$= {}^{11}C_3 = \frac{11 \times 10 \times 9}{3 \times 2} = 165$$

10. (b)

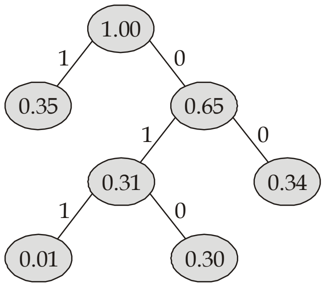
O(V) → Time taken to build the vertices min heap.

O(V logV) → Extract minimum vertex and heapify.

O(E logV) → For each updates have to be mode in the min heap before next extraction.

11. (b)

Using min heap data structure:



Expected length: [0.35 × 1 + 0.34 × 2 + 0.30 × 3 + 0.01 × 3] × 200

= [0.35 + 0.68 + 0.90 + 0.03] × 200

= [1.96] × 200

= 392

12. (a)

Since by looking through options, we get to know 'b' will be the start vertex.

	a	b	c	d	e	f	g	h
	∞	0	∞	∞	∞	∞	∞	∞
b	5	–	∞	1	3	∞	∞	∞
(b – d) d	5	–	∞	–	3	6	∞	∞
(d – e) e	5	–	4	–	–	6	9	∞
(e – c) c	4	–	–	–	–	6	9	7
(c – a) a	–	–	–	–	–	6	9	7
(d – f) f	–	–	–	–	–	–	9	7
(c – h) h	–	–	–	–	–	–	9	–
(e – g) g	–	–	–	–	–	–	–	–

So, correct sequence will be (b – d), (b – e), (e – c), (c – a), (d – f), (c – h), (e – g).

13. (d)

Property of the output of each pass of quicksort:

I. The pivot elements is on its correct position.

II. All the elements on the left of the pivot are smaller than it and elements on the right are greater than it.

If we consider the sorted permutation of the above sequence it will be, 6, 9, 11, 13, 15, 17, 18, 24.
Comparing both it can be observed that elements 11, 13 and 24 are on their correct positions.
Besides that they are satisfying II property also.

Hence, possible pivots, 11, 13, 24

Sum, 11 + 13 + 24 = 48

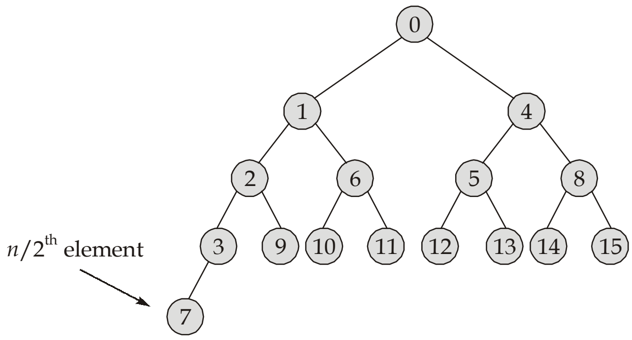
14. (b)

Since there are total 1024 elements hence there will be total 11 levels of the heap. $(n/2)^{th}$ element can be present at last level in worst case i.e. 11^{th} level.

$(n/2)^{th}$ element can also be present in best case level i.e. 2^{nd} level

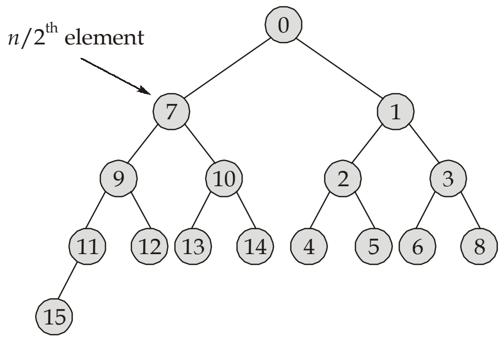
Assume for n = 16

In worst case:



$(n/2)^{th}$ element is at last level.

In best case:



$(n/2)^{th}$ element is at second level.

So, difference = [11 – 2] = 9

15. (c)

Time complexity to sort n elements using merge sort = $\Theta(n\log n)$

$$\Theta(n) = \Theta\left(\frac{n}{\log n}\log\frac{n}{\log n}\right)$$

$$\Theta(n) = \Theta\left(\frac{n}{\log n}[\log n - \log\log n]\right)$$

$$\Theta(n) = \Theta\left(\frac{n}{\log n}\log n\right) \quad [\log n - \log\log n = O(\log n)]$$

$$\Theta(n) = \Theta(n)$$

16. (d)

A. Matrix chain multiplication : (n^3)

B. Travelling salesman problem : (n^n)

C. 0/1 knapsack : (mn)

D. Fibonacci series : $O(n)$

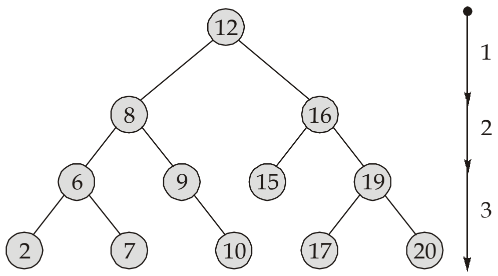
17. (d)

Post order: 2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12

Inorder of BST must be sorted order:

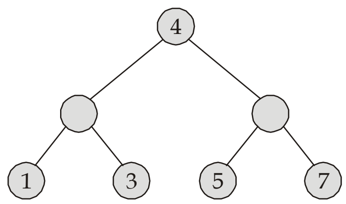
2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20

So, tree will be:



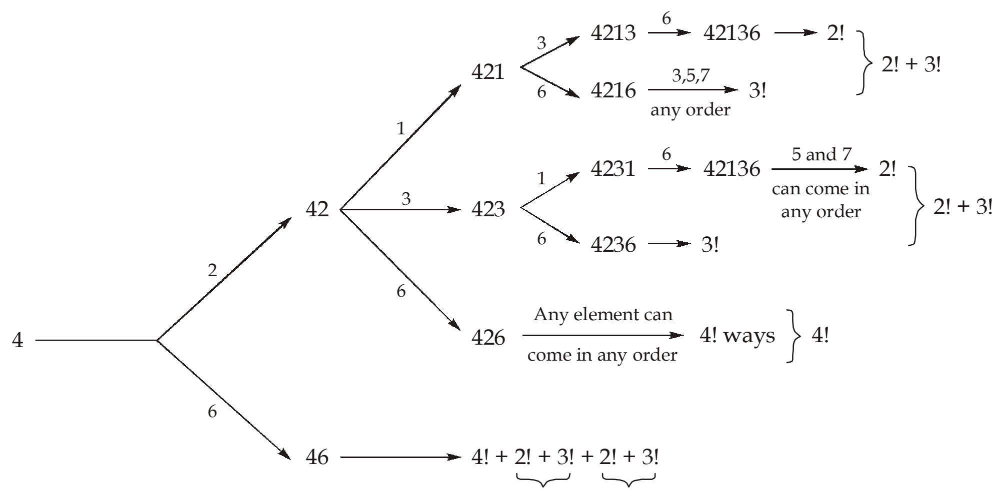
18. (b)

Since 4 is root element, so



(1, 3, 5, 7) can be inserted in any order since these are leaf nodes. However, 6 needs to be inserted before 5 and 7 and 2 needs to be inserted before 1 and 3.

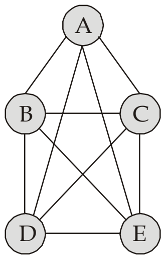
4 being the root node, needs to be inserted first of all.



$$
\begin{aligned}
\text{Total possibilities} &= 2(4! + 2(2! + 3!)) \\
&= 2(24 + 2(2 + 6)) \\
&= 2(24 + 16) = 2(40) \\
&= 80 \text{ ways}
\end{aligned}
$$

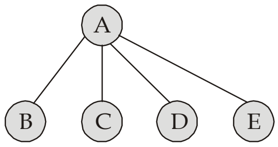
19. (c)

Complete graph on 5 vertices:



BFS traversal:

1. We have 5 choices to select initial vertex i.e. A, B, C, D and E.
2. With initial vertex as A



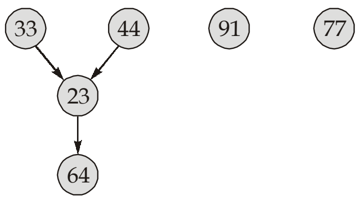
= 4! traversal

So total number of traversal are 5 × 4! = 5! = 120

Note: Number of BFS traversals on complete graph with n vertex are n!

20. (b)

Here some of the dependencies are presents:



So, number of possibilities are:

1. 2 choices for 33 and 44 either 33 then 44 or 44 then 33.
2. After that 23 will be come.
3. After that 64 will come.

Now, here 91 and 77 can come in any order i.e. 5 × 6

So, total choices will be = 2 × 5 × 6
= 60

21. (d)

In best case merge sort time complexity = O(n log$n$)

For quick sort in best case = O(n log$n$)

For selection sort = O(n^2)

For insertion sort = O(n)

22. (b)

If $\forall i\ \forall j\ (P(i \to j) \wedge P(j \to i))$ then i and j belongs to same strongly connected components where i and j are vertices and $P(i \to j)$ means there is a path from i to j.

Solving by elimination

Option (d): P(U → T) does not exists thus false.

Option (c): From R we can reach P Q V S T and from P Q V S T we can reach R thus R should not be separated thus false.

23. (b)

We know the range of the element present in the array A, so can use counting sort also, takes O(n) time.

$$C \cdot n = 650$$
$$C \cdot 50 = 650$$
$$C = 13$$

For $n = 300$

$$300 \times 13 = 3900$$

24. (d)

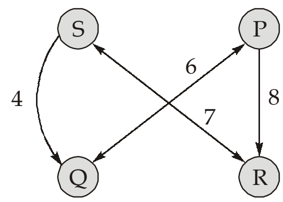
(ii) choosing median element as pivot divide the array into two equal half and time complexity O($n$log$n$).

For (i) and (iii) choosing pivot randomly or middle element does not guaranty O($n$log$n$) time complexity.

25. (c)

A : Quick sort uses divide and conquer.
B : Longest common subsequence uses dynamic programming.
C : Kruskal's algorithm is greedy algorithm.
D : Shortest distance from a given node to every node is greedy algorithm.

26. (b)



Optimal cost = 4 + 6 + 7 + 8 = 25

27. (b)

p = "abcbdab"
q = "bdcabc"
LCS (p, q) = abcbdab, bdcabc
First LCS = "bcab"
Second LCS = "bdab"

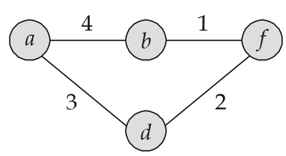
Then total 2 LCS.

28. (c)

In P R U V W S Q T is not DFS traversal because after U, V can not come.

29. (a)

(a) For a directed graph if DFS tree does not have back edges then there is no cycle.
(b) Shortest path between two vertices may not be unique.



Shortest path between $a - f$ is not unique.

(c) A complete graph can have maximum $n^{n-2} = 4^{4-2} = 16$ MST.

So only option (a) is true.

30. **(d)**

0		
1	23	→ 34
2	24	
3	36	
4		
5	49	→ 46 → 60
6		
7		
8		
9	20	
10		

If the new key is inserted in empty slot 6, total 11 slot.

$$\text{Probability} = \frac{6}{11} = 0.54$$

■■■■