CLASS TEST

S.No.: 02SKCS_GHI_18122023

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

ALGORITHMS

COMPUTER SCIENCE & IT

Date of Test : 18/12/2023

ANSWER KEY

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

DETAILED EXPLANATIONS

1. (b)

Size m $\Rightarrow$ To delete root: O (log n)

Size $\sqrt{n}$ $\Rightarrow$ To delete root: $O(\log\sqrt{n}) = O\left(\log n^{1/2}\right) = O\left(\frac{1}{2}\log n\right) = O$ (log n)

2. (b)

Prim's algorithm will pick up the edge with least weight for a particular node, [provided it does not form a cycle] weight of edge (V_{i-1}, V_i) or $(V_i, V_{i-1}) = 1$

$\therefore$ MST will be

$$V_1 \underset{1}{\text{——}} V_2 \underset{1}{\text{——}} V_3 \underset{1}{\text{——}} V_4 \cdots V_n$$

$\therefore$ Total edge weight = $2 \times (n - 1) = 2n - 2$.

3. (d)

In worst case, i.e., dense graph, $E = O(V^2)$

$\therefore$ Time complexities of

1. 'V' invocations of Dijkshtra algo = $O(V^3 \log V)$
2. 'V' invocations of Bellman-ford algo = $O(V^4)$
3. '1' invocation of Floyd-warshall algo = $O(V^3)$

$\therefore$ Floyd-warshall is best.

4. (a)

Since two arrays are already in sorted order. We first select median of X and Y array in O(1) time lets say 'a' and 'b' respectively. Compare X[a] and Y[b] if X[a] < = Y(b) then we apply above procedure to X[a,...n], Y[b,...,n] till their are ≤ 2 elements left in each array.

So, total time = log(n) + O(1) = log (n).

5. (a)

We need to delete element from the array and insert into the AVL tree. Deletion from the array satisfying heap property will take $O(\log n)$ time for each element.

$$\begin{aligned}\text{Total time} &= O(n \log n) + O(n \log n) \\ &= O(n \log n)\end{aligned}$$

Note: $O(n \log n)$ time to satisfy the heap property of the array for every deletion and $O(n \log n)$ time to construct the AVL tree.

6. (a)

Dijkstra's algorithm will output same as breadth first search on graph and will take $O(m + n)$ time.

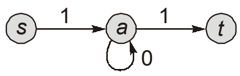
7. (c)

Apply BFS algorithm on graph will give number of nodes at height "$v/2$" from particular node it will take $O(v + e)$ time.

8. (a)

- In Radix Sort, all input orderings give the worst-case running time, the running time does not depend on the order of the inputs in any significant way.
- The parent pointers may not lead back to the source node if a zero length cycle exists.
  In the example below, relaxing the (s, a) edge will set $d[a] = 1$ and $\pi[a] = s$. Then, relaxing the (a, a) edge will set $d[a] = 1$ and $\pi[a] = a$

Following the π pointers from t will no longer give a depth to s, so the algorithm is incorrect.



9. (d)

A. Matrix chain multiplication : (n^3)

B. Travelling salesman problem : (n^n)

C. 0/1 knapsack : (mn)

D. Fibonacci series : $O(n)$

10. (c)

Finding minimum edge each time will take (v) time, this will done for $(V - 1)$ time. Will take worst case time

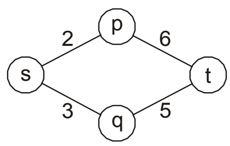
$$= O(|V|) \times (V - 1)$$
$$= O(|V|^2)$$

11. (d)

1. If the graph has a Hamiltonian cycle, then it is possible depending on the graph ordering of the graph, that DFS will find that cycle and that the DFS tree will have depth V-1. However, DFS is not guaranteed to find that cycle. If DFS does not find that cycle then the depth of the DFS tree will be less than V-1.
2. False because, if the graph contains a negative-weight cycle, then no shortest path exist.

13. (b)

Even if no two edges have the same weight, there could be two paths having the same weight.



To find the shortest path between s and t.

$$s \to p \to t \Rightarrow 8$$
$$s \to q \to t \Rightarrow 8$$

So, there are two distinct paths.

14. (d)

Number of leaf nodes = n

Let internal nodes be K

∴ Total nodes = K + n

For m-ary tree, number of leaf nodes with K internal nodes = (m – 1) K + 1

∴ (m – 1) K + 1 = n

$$\therefore \qquad K = \frac{n-1}{m-1}$$

∴ Total number of nodes

$$= \frac{n-1}{m-1}+n = \frac{n-1+n(m-1)}{m-1} = \frac{n-1+nm-n}{m-1} = \frac{nm-1}{m-1}$$

15. (b)

Considering each statement :

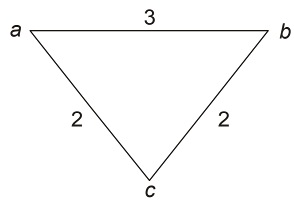
S_1 :It takes O(1) time to delete the node but, the heap property gets destroyed, so to resume the property of a heap, time taken will be O(log$n$).

S_2 :Fractional Knapsack using greedy approach takes O(n log$n$) time complexity.

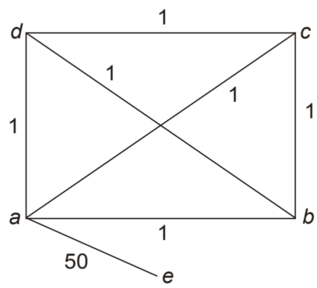
S_3 :Selection sort algorithm is designed using Greddy approach.

16. (d)

I. MST contain *ac* and *bc* but not contain *ab*, which is the shortest path between *a* and *b*



II. We may be forced to select the edges with weight much higher than average



$$\text{Average weight} = \frac{50+6}{7} = 8$$

$$\text{Expected MST weight} = 4 \times 8 = 32$$

$$\text{Actual MST weight} = 50 + 6 = 56$$

17. (a)

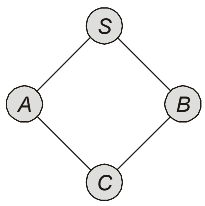
Starting vertex is D

	A	B	C	D	E	F	G	H
D	∞	∞	∞	0 Nill	∞	∞	∞	∞
F	16	17_D	13_D	–	∞	3_D	∞	8_D
B	16	7_E	13_D	–	∞	–	∞	8_D
H	16	–	13_D	–	∞	–	12_B	8_D
E	16	–	13_D	–	10_H	–	12_B	
G	16	–	13_D	–	–	–	12_B	
C	16	–	–	–	–	–	–	–
A	–	–	–	–	–	–	–	–

So the order of relaxed the vertices by using Dijkastra's algorithm is DFBHEGCA.

18. (c)

Consider a graph



During the breadth first traversal of the graph. The status of the queue will be as follows :

S	

A	B

$\therefore$ $S - A \rightarrow$ 1 edge

$S - B \rightarrow$ 1 edge

Difference = 0

B	C

$\therefore$ $S - B \rightarrow$ 1 edge

$S - C \rightarrow$ 2 edge

Difference = 1

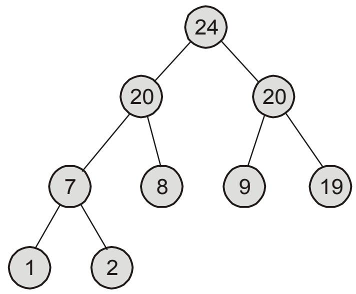
Hence, statements S_1 and S_3 are correct.

19. (c)

$$f(n) = \sum_{i=0}^{99} \sum_{j=0}^{n-1} \left(\sum_{k=0}^{j-1} 1 + \sum_{m=0}^{i-1} 1 \right) = \sum_{i=0}^{99} \sum_{j=0}^{n-1} \sum_{k=0}^{j-1} 1 + \sum_{i=0}^{99} \sum_{j=0}^{n-1} \sum_{m=0}^{i-1} 1 = O(n^2)+O(n)=O(n^2)$$

20. (c)

The given heap is



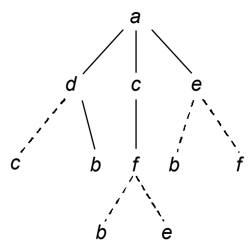
When we delete root element in max heap we replace root with last node in the tree. Now, while Max-Heapify down, we select the maximum of the two children and swap it with root.

Here the two children are both 20 and 20 and we can swap with either of them and the only two possible heaps after deletion and Heapifying are

1. 20, 8, 20, 7, 2, 9, 19, 1
2. 20, 20, 19, 7, 8, 9, 2, 1

21. (b)

Given BFS traversal on G is :



Tree edges are: $\{a, d\}, \{a, c\}, \{a, e\}, \{d, b\}, \{c, f\}$

Cross edges are: $\{d, c\}, \{e, b\}, \{e, f\}, \{f, b\}, \{f, e\}$

$\Rightarrow$ $\{d, b\}$ is not a cross edge.

22. (b)

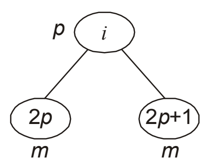
A	27	12	45	31	23	26
	1	2	3	4	5	6

A has following inversions:

$(i, j) \Rightarrow (1, 2), (1, 5), (1, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)$

If $i < j$ and $A[i] > A[j]$ then (i, j) is inversion pair [for $1 \le \{i, j\} \le n$]

23. (b)



{If $A[2p + 1] > A[2p]$ then $m = 2p + 1$

else $m = 2p$}

X is : $2p \le n$

Y is : $(2p + 1) \le n$

Z is : $A[2p + 1] > A[2p]$

24. (b)

$$W = 8 \text{ (capacity)}$$

Feasible solutions:

(i) $\{I_1, I_3, I_4\}$,

(ii) $\{I_2, I_3\}$

Profit of $\{I_1, I_3, I_4\}$ is 23

profit of $\{I_2, I_3\}$ is 15

Optimal solution is $\{I_1, I_3, I_4\}$ with capacity of 8 and maximum profit 23 produced.

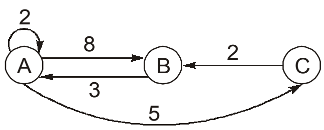
$\therefore$ I_2 is not selected in the solution.

25. (d)

ABDE H FGH is not DFS traversal

Traversal from node D to E fail the DFS algorithm.

26. (c)



$$\begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 2 & 8 & 5 \\ 3 & \infty & \infty \\ \infty & 2 & \infty \end{bmatrix} \end{array}$$ is adjacency matrix for above graph.

$$A_0 = \begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & \infty \\ \infty & 2 & \infty \end{bmatrix} \end{array} \Rightarrow A_1 = \begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & 8 \\ \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$A_2 = \begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & 8 \\ 5 & 2 & 0 \end{bmatrix} \end{array} \Rightarrow A_3 = \begin{array}{cc} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & 8 \\ 5 & 2 & 0 \end{bmatrix} \end{array}$$

27. (d)

Longest common subsequence, longest increasing subsequence, sum of subsets, optimal BST, 0/1 knapsack, matrix chain multiplication, Travelling salesperson, Balanced partition, Fibonacci sequence, Multistage graph problems are solved by using dynamic programming.

28. (c)

Only $j - 1$ at $\boxed{X}$ and $i - 1$ at $\boxed{Y}$ gives the correct implementation of insertion sort.

29. (c)

$$f(n) = 3n + 100 = \theta(n)$$
$$g(n) = n + \log n = \theta(n)$$
$$\therefore \quad f(n) = \theta(g(n)) \text{ is correct}$$

30. (b)

	A	B	C	D	E	F	G	H
	$\underset{Nill}{0}$	∞	∞	∞	∞	∞	∞	∞
A	–	$\underset{A}{2}$	$\underset{A}{15}$	$\underset{A}{23}$	∞	∞	∞	∞
B	–	–	$\underset{A}{15}$	$\underset{A}{23}$	$\underset{B}{17}$	$\underset{B}{7}$	∞	∞
F	–	–	$\underset{F}{13}$	$\underset{A}{23}$	$\underset{F}{11}$	–	$\underset{F}{36}$	$\underset{F}{y+7}$
Given $\rightarrow E$	–	–	$\underset{F}{13}$	$\underset{A}{23}$	–	–	$\underset{F}{36}$	$\underset{E}{x+11}$

Compare above table with given table

$$x + 11 = 19$$
$$x = 8$$
$$y + 7 \geq x + 11$$
$$y + 7 \geq 8 + 11$$
$$\Rightarrow \quad y \geq 12$$

■■■■