OVERALL ANALYSIS | Solution Report

All | Correct Answers | Wrong Answers | Not Attempted Questions

Q.1) Max Marks: 1

What is the time complexity of the following code

```
sum=0;
for(i=n;i>=0;i--)
{
      for(j=1;j<n;j=j*2)
      {
            sum+=(j+i);
      }

}
```

A **O(logn)**

B **O(m log n)**

C $O(n^2)$

D **O(n log n)** Correct Option

Solution: (D)
The outer loop is executed n times, and the inner loop is executed log n times as at each iteration j is multiplied by 2 and each time, it will require O(log n) steps to reach n. Therefore total time complexity O(n log n)

Q.2) Max Marks: 1

The time complexity of an algorithm following the given recurrence relation is

T(n)=5T(n/3)+log n

A **O(1)**

B **O(n log n)**

C $O(\sqrt{n})$

D $O(n^2)$ Correct Option

Solution: (D)
On solving the recurrence relation using masters theorem a=5 b=3 k=0 we get $T(n)=O(n^{\log_3 5})$
The closes among all the options is D.

Q.3) Max Marks: 1

The worst case running time to search for an element in a balanced in a binary search tree with n*2^n elements is:

A **Θ(n log n)**

B **Θ(n 2^n)**

C **Θ(n)** Correct Option

Solution: (C)
Solution: Ans is (C)Θ(n)

Balanced binary tree means AVL tree then:

Time taken to search an element is Θ(log n) where n is no. of elements in AVL tree.As number of elements given is n*2^n, the searching complexity will be Θ(log(n*2^n)) which can be written as:

= Θ(log (n*2^n))

=Θ((log n) + Θ(log (2^n)))

= Θ((log n) + Θ(n log (2)))

= Θ((log n) + Θ(n))

As log n is asymptotically smaller than n, Θ(log(n)) + Θ(n) can be written as Θ(n)

which matches option C.

D **Θ(log n)**

Q.4) Max Marks: 1

An AVL tree of height h=4 with minimum nodes is given then find the number of leaf nodes in this tree.

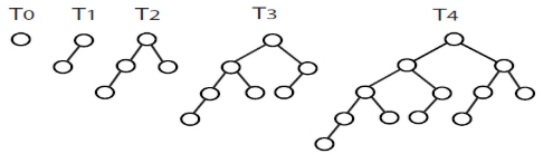
A **h**

B **h-1**

C **h+1** Correct Option

Solution: (C)

Solution: Ans is (C) h+1
AVL tree are height balanced tree. In these trees height balanced factor for nodes can be 0, 1, +1 so height balanced trees will be like this.



By the above figures of AVL tree, of different heights we can check that. In the AVL tree of height 4, there are total 5 leaf nodes. So answer will be h+1.

D **h+2**

Q.5) Max Marks: 1

The time complexity to topologically sort the vertices of a given graph G with n vertices and m edges is given by

A **O(n log m)**

B **O(nm)**

C **O(n+m)** Correct Option

Solution: (C)
Topological sorting of the vertices can be done using DFS and additional linked list to maintain the topological order, once a particular vertex is explored in DFS it is added to the first node in the linked list once the DFS of the graph is completed the linked list gives us the topological ordering. Time for DFS =O(n+m) and Time for inserting into linked list is O(n), total time required=O(n+m).

D **None of the above.**

Q.6) Max Marks: 1

Consider the following segment of code

```
for(i=0;i<2*n;i=i+n)
{
      for(j=1;j<n;j=j*2)
      {
            printf("%d%d",i,i);
      }
      printf("\n");
}
```

What is the number of new lines printed by the above fragment of code

A **O(log n)**

B **O(n)**

C **O(n logn)**

D **None of the above.** Correct Option

Solution: (D)
The no of new lines are only printed by the printf statement of the outer for loop. the outer loop only executes 2 times which is O(1) times.

Q.7) Max Marks: 1

Which of the following sorting algorithm is not an in-place sorting technique

A **Heap sort**

B **Merge sort** Correct Option

Solution: (B)
Heapsort, Bubble Sort and Merge sort do not require additional space and hence are in-place sorting algorithms whereas Mergesort requires additional space, therefore, Merge Sort is not an in-place sorting algorithm.

C **Bubble sort**

D **Insertion sort.**

Q.8) Max Marks: 1

How many queues are required for implementing priority queue

A **only 1**

B **2** Correct Option

Solution: (B)
Solution: Ans is (B) 2
Priority queue has the priority associated with each element. So 1 queue will store the elements and 2nd queue will give priority to the elements which are stored in the queue. We will need 2-queues.

C **3**

D **4**

Q.9) Max Marks: 1

From the given in-order of balanced binary search tree 9, 12, 14, 17, 19, 22, 28, 32, 39, 55, 58 left subtree and right subtree with equal nodes and height 2. And pre-order of this is 22, 14, 12, 9, 17, 19, 39, 32, 28, 55, 58. Then find out sum of all leaf nodes.

A **56**

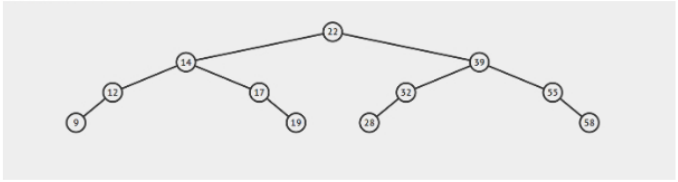
B **58**

C **86**

D **114** Correct Option

Solution: (D)
Solution: Ans is (D) 114
Here, we are starting with in-order and pre-order
In-order: 9, 12, 14, 17, 19, 22, 28, 32, 39, 55, 58
Pre-order: 22, 14, 12, 9, 17, 19, 39, 32, 28, 55, 58

As we know that in-order traversal is LEFT, ROOT, RIGHT while pre-order traversal is ROOT, LEFT, RIGHT. So 22 will be root node and 14 will be the first node at left subtree. Then construct tree by knowing order of node in in-order traversal and pick the element from the pre-order traversal. So final tree will be like this.



So leaf nodes will be 9,19,28,58 and sum of these elements will be 114.

Q.10) Max Marks: 1

Find the sum of all the PRIME numbers of the queue after performing all operations.
enqueue(4)enqueue(5)dequeue()enqueue(8)enqueue(9)enqueue(6)dequeue()enqueue(7)
dequeue()enqueue(1)enqueue(2)enqueue(3)dequeue()

A **10**

B **11**

C 12 Correct Option

Solution: (C)

Solution: Ans is (C) 12

Here, when we go according to question then we need to take a queue and we know queue works on first in first out concept so

4	5			

dequeue (): it will delete 4 from the queue

	5	8	9	6

dequeue (): it will delete 5 from the queue

		8	9	6	7

dequeue (): it will delete 8 from the queue

			9	6	7	1	2	3

dequeue (): it will delete 9 from the queue

				6	7	1	2	3

And prime numbers are 2, 3, 7 and the sum of these three is 12 only.

D 13

Q.11) Max Marks: 2

Consider a Circular Queue 'q' is implemented whose size is given 15 and the locations of elements of the queue are q[0], q[1], q[2].....,q[14]. In this given circular queue currently rear is pointing at q[7] after inserting 11th element, then find the position of front in this queue (as element insertion can be started from anywhere and it will be moved forward in ascending order like (0,1,2,3----,13,14,0,1).

A q [2]

B q [3]

C q [11]

D q [12] Correct Option

Solution: (D)

Solution: Ans is (D). q [12]

If we start insertion from location q[12].

Then q[7] = 11th element

q[6] = 10th element

q[5] = 9th element

q[4] = 8th element

q[3]= 7th element

q[2] = 6th element

q[1] = 5th element

q[0] = 4th element

q[14]= 3rd element

q[13] = 2nd element

q[12] = 1st element

Q.12) Max Marks: 2

Given the following set of jobs to be scheduled on a single processor machine and their deadlines and profit, what is the optimum profit which can be obtained by any schedule___

Job	Finish Time (Deadline)	Profit
1	4	20
2	3	10
3	2	12
4	5	17
5	5	12
6	2	14
7	3	15

Correct Answer

Solution: (78)

For job sequencing within deadlines, we have to schedule the most profitable jobs as late as possible here is the optimum schedule.

Time	1	2	3	4	5
Job	3 or 5	6	7	1	4
Profit	12	14	15	20	17

Total profit =78

Q.13) Max Marks: 2

The number of comparisons in the bubble sort sorting is given by

A $O(n^2)$ Correct Option

Solution: (A)

In bubble sort the comparisons are in the following pattern,

1st iteration - n-1 comparisons
2nd iteration n-2 comparisons
3rd iteration n-3 comparisons
....
N-1 th iteration 1 comparison,
Total no of comparisons= $n(n-1)/2 = O(n^2)$

B **O(n)**

C **O(n logn)**

D **None of the above.**

Q.14) Max Marks: 2

Given string "gateappliedcourse" is stored in the stack character by character. When we get null character then we start popping characters from stack and enqueued into modified queue. Then what will be the output of modified queue for dequeued characters if it is working with three functions enqueue, reverse, dequeue.(Assume it is reversing all the elements before performing dequeuer operation)

A **esruocdeilppaetag**

B **gateappliedcourse** Correct Option

Solution: (B)

Solution: Ans is (B) gateappliedcourse

Here, we need to remember two concepts stack- LIFO and queue- FIFO
First, we are inserting string characters one by one into the stack and after the end of string when we get null character. Null character is always found at the end of string. So after pushing string characters in the stack we get null characters.

G	a	t	e	a	p	p	l	l	e	d	c	o	u	r	s	e

Then characters are removed from the stack and inserted in the modified queue. After insertion of characters, it is performing reverse operation so characters are reversed and at the time of dequeue it will simply print the same string as characters will be printed in the same order.

C **gatedeilppacourse**

D **etagappliedesruoc**

Q.15) Max Marks: 2

If we are hashing given array elements as h(k)mod 9 in the hash table of 9 slots starting from zero index element and collisions are resolved using linear probing then find out which element of highest array index(given) is inserted without collision. Array is given as follows:

Index	0	1	2	3	4	5	6	7
keys	26	2	5	9	14	38	49	87

Choose the correct option for the above scenario.

A. 87
B. 49
C. 38
D. 9

A **87**

B 49 Correct Option

Solution: (B)

Solution: Ans is (B)49

For inserting elements we are using h(k)mod 9, hash table is given as follows:

0	9
1	
2	2
3	38
4	49
5	5
6	14
7	87
8	26

26mod 9 = at 8th place in hash table
2mod 9 = at 2nd place in hash table
5mod 9 = at 5th place in hash table
9mod 9 = at 0th place in hash table
14mod 9 = at 5th place but it's not empty then at 6th place
38mod 9 = at 2nd place but it's not empty then at 3rd place.
49mod 9 = at 4th place
87mod 9 = at 6th place but it's not empty then at 7th place

By this we can conclude, element 49 which has highest array index is inserted without collision in hash table.

C 38

D 9

Q.16) Max Marks: 2

What is the length of the longest common subsequence for the given strings is____
"abracadabra"
"abcdefabcdef"

Correct Answer

Solution: (6)
The longest common subsequence of the two strings is "abcdab"

Q.17) Max Marks: 2

There is a queue q with enqueue, dequeue operations, which already has 8 elements with it as 4, 5, 6, 9, 1, 2, 3, 7 (4 is enqueued first and 7 at last). Then run the following given code. Here, s is the stack with push and pop operations.

```
for ( i=1; i<=5; i++)
{
x = q.dequeue();
s.push(x);
}
 for ( i=1;  i< = 3; i++)
{
 x= s.pop();
enqueue(x);
}
```

Find the order of elements after performing all operations.

A 4, 5, 6, 7, 3, 2

B 4, 5, 6, 9, 1, 2

C 2, 3, 7, 4, 5, 6

D 2, 3, 7, 1, 9, 6 Correct Option

Solution: (D)
Solution: Ans is (D) 2, 3, 7, 1, 9, 6
In the starting q have elements as
Q = 4, 5, 6, 9, 1, 2, 3, 7
After performing dequeue operation 5 times as loop is given up to 5, remaining Q will be as Q = 2, 3, 7
And stack s will be s = 4, 5, 6, 9, 1 (top)
Then for the second for loop, which will be executed three times, so the remaining elements in the stack s = 4, 5
And popped elements are 1, 9, 6 which we enqueue in the q
Then order of elements in the q after inserting these elements is 2, 3, 7, 1, 9, 6
So option d is correct

So option d is correct.

Q.18) Max Marks: 2

Consider the following sorting algorithm if "A" represents the array to be sorted and i, j represent the index to the first and the last element of the array, the time complexity of the such an algorithm is

```
STOOGE-SORT(A, i, j)
1. if A[i] > A[j]
2.     then exchange A[i] ↔ A[j]
3. if i + 1 ≥ j
4.     then return
5. k ← ⌊(j − i + 1)/3⌋
6. STOOGE-SORT(A, i, j − k)
7. STOOGE-SORT(A, i + k, j)
8. STOOGE-SORT(A, i, j − k)
```

A $O(n^{(\log 3/\log 1.5)})$ — Correct Option

Solution: (A)

The recurrence relation for the above can be given by
T(n)=3(2n/3)+1
On solving using Masters Theorem or by recursion tree method we get $O(n^{(\log 3/\log 1.5)})$.

B $O(n^{(\log 2)})$

C $O(n^2)$

D $O(n^{(\log 1.5/\log 3)})$

Q.19) Max Marks: 2

Consider the following singly linked-list.

P = A(2000) → B(3000) → C(4000) → D(5000) → E(6000) → F(7000) → G (NULL)

And then after executing following four statement

(I). struct node *Q;

(II). Q = P → next → next → next

(III). P → next → next → next → next → next = Q → next → next

(IV). printf("%d", Q → next);

Choose the correct option for the above scenario

A **D**

B **68**

C **E**

D **69** — Correct Option

Solution: (D)

Solution: Ans is (D) 69

(I). struct node *Q; // initializing one more variable for storing linked list

(II). Q = P → next → next → next // start pointing at 4000

(III). P → next → next → next → next → next // start pointing at 5000 and 5000 ->next

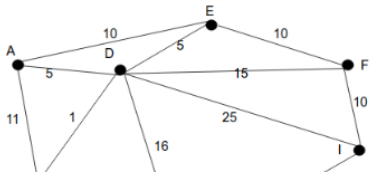
= Q → next → next // Q was pointing at 4000 here also 5000 -> next

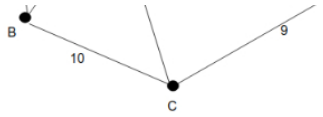
(IV). printf("%d", Q → next); // we will print content of 5000 ->next i.e. E and its ascii value is 69

So answer (C) E is true.

Q.20) Max Marks: 2

The number of minimal spanning trees for the given graph below are _____

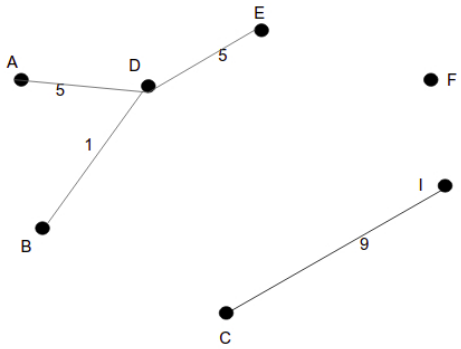




Correct Answer

Solution: (3)

If we apply Kruskal's algorithm for MST we get the following graph till the weight of the edges<=9



Now we have 4 edges of weight 10 AE (this forms a cycle so it is ruled out), now we have (EF), (FI) and BD adding any two edges will make it a MST this can be done in C(3,2) ways. Therefore the no of minimal spanning tree is 3.

close