Nitish Kumar Gupta
Course: GATE
Computer Science Engineering(CS)

HOME
MY TEST
BOOKMARKS
MY PROFILE
REPORTS
BUY PACKAGE
ASK AN EXPERT
OFFER
EXCLUSIVE OFFER FOR OTS STUDENTS ONLY ON BOOK PACKAGES

SINGLE SUBJECT :ALGORITHMS (GATE - 2019) - REPORTS

OVERALL ANALYSIS | COMPARISON REPORT | SOLUTION REPORT

ALL(33) | CORRECT(11) | INCORRECT(7) | SKIPPED(15)

Q. 1

What data structure would you choose to implement "undo" feature in a word processor?

Have any Doubt ?

A. Queue

B. Dequeue

C. Stack — Correct Option

Solution :
(c)
Stack is most suitable data structure since it works in LIFO ordering. All the actions are pushed into stack and whatever action has to be reversed (undoed) can be just popped off the top of stack.

D. None of the above

QUESTION ANALYTICS +

Q. 2

Which of the following tree traversal technique is not based on depth first search?

Have any Doubt ?

A. Preorder traversal

B. Level order traversal — Your answer is **Correct**

Solution :
(b)
Level order traversal also known as breadth first search while other three are depth first search.

C. Postorder traversal

D. Inorder traversal

QUESTION ANALYTICS +

Q. 3

Consider the following statement:
S_1 : Ternary search is more efficient than binary search for finding an element in a sorted array.
S_2 : Worst case deletion and insertion in an AVL tree has same complexity.

FAQ | Have any Doubt ?

A. S_1 is correct, S_2 is not

B. S_2 is correct, S_1 is not — Your answer is **Correct**

Solution :
(b)

- Recurrence relation for ternary search: $T(n) = T(n/3) + 4$, $T(1) = 1$
  Recurrence relation for binary search: $T(n) = T(n/2) + 2$, $T(1) = 1$

	Number of comparisons
Binary Search:	$2\log_2 n + 1$
Ternary Search:	$4\log_3 n + 1$

  Since $(2\log_2 n + 1) < (4\log_3 n + 1)$. Hence binary search is preferred.
- In AVL tree, worst case deletion and insertion both have O(log$n$) complexity

C. Both are correct

D. Both are incorrect

QUESTION ANALYTICS +

Q. 4

Match **List-I (Solution approach)** with **List-II (Problems)** and select the correct answer using the codes given below the lists:

List-I	List-II
A. Divide and Conquer Approach	1. Graph coloring
B. Dynamic programming	2. 0/1 Knapsack
C. Greedy Approach	3. Strassen Matrix Multiplication
D. Backtracking	4. Huffman Encoding

Codes:

	A	B	C	D
(a)	3	1	2	4

(a)	3	1	2	4
(b)	3	2	1	4
(c)	3	2	4	1
(d)	4	3	1	2

Have any Doubt ?

A a

B b

C c — Your answer is **Correct**

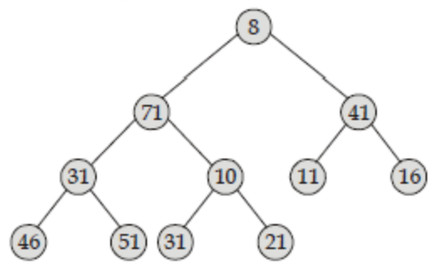
Solution :
(c)
Divide and Conquer Approach - Strassen Matrix Multiplication
Dynamic programming - 0/1 Knapsack
Greedy Approach - Huffman Encoding
Backtracking - Graph coloring

D d

QUESTION ANALYTICS +

Q. 5

A min-max heap has the following property: "A almost complete binary tree where each node at an even level in the tree is less than all of its descendants and each of the node at an odd level in the tree is greater than all of its descendants".



What is the worst case time complexity of "deleting minimum element" and "deleting maximum element" on above data structure?

Have any Doubt ?

A O(log*n*) and O(log*n*) — Correct Option

Solution :
(a)
deletemin( ): Root element is the smallest element. Remove it and swap it with the rightmost leaf node to maintain heap property. Then search for the minimum element out of the remaining elements by looking in the lowermost min (even) level. Let it be at index '*m*'. Swap this element at index m with the root. Do this recursively for the subtree rooted at index m to maintain the min-max property.
deletemax( ): The max element is one out of the two children of root. Find it and then follow the same steps as deletemin( ) operation, this time choosing the max element out of the remaining to be swapped at every stage.

B O(*n*) and O(log*n*)

C O(log*n*) and O(*n*)

D O(*n*) and O(*n*)

QUESTION ANALYTICS +

Q. 6

Consider the recurrence relation:

$$T(n) = \begin{cases} T(\sqrt{n}) + \log n & \text{if } n \geq 2 \\ O(1) & \text{else} \end{cases}$$

Then T(*n*) in terms of Θ notation is

Have any Doubt ?

A Θ(log*n*) — Correct Option

Solution :
(a)

$$T(n) = T(\sqrt{n}) + \log n$$

Put, $n = 2^m$

$$T(2^m) = T(2^{m/2}) + m$$

Put, $T(2 = S($

So, $S(m) = S\left(\frac{m}{2}\right) + m$

By applying Master theorem:
We get, $S(m) = \Theta(m)$
Put value of $m = \log_2 n$ [$2^m = n$, $m = \log_2 n$]
So, $T(n) = \Theta(\log n)$

B Θ(*n*)

C Θ(log log*n*)

D Θ(*n*log*n*)

QUESTION ANALYTICS +

Q. 7

Consider we have given two large integers in binary format each of length *n* bits. What is the worst case time complexity to find product of two integers?

Have any Doubt ?

A $O(\log n)$

B $O(n)$

C $O(n^{1.59})$ Correct Option

Solution :
(c)
The algorithm is known as Karatsuba algorithm and is based on divide and conquer strategy. Recurrence relation is

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$

Solution of this recurrence is $O(n^{1.59})$.

D $O(n^2)$ Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 8

Which of the following is correct?

Have any Doubt ?

A $\frac{e^{n\log n}}{n} < 2^{n\log n} < n^{\sqrt{n}}$

B $2^{n\log n} < \frac{e^{n\log n}}{n} < n^{\sqrt{n}}$

C $n^{\sqrt{n}} < \frac{e^{n\log n}}{n} < 2^{n\log n}$ Your answer is **Correct**

Solution :
(c)

1. $\frac{e^{n\log n}}{n} = e^{n\log n - \log n}$
2. $2^{n\log n} = e^{(\log_2)^{n\log n}} = (e^{n\log n})$
3. $n^{\sqrt{n}} = (e^{\log n})^{\sqrt{n}} = (e^{\sqrt{n}\log n})$

D $2^{n\log n} < n^{\sqrt{n}} < \frac{e^{n\log n}}{n}$

QUESTION ANALYTICS +

Q. 9

What is the time complexity of the following program?

```
void fun(int n) {
    int i, j, counter = 0;
    for (i = 1; i ≤ n; i++) {
        for (j = 1; j × j ≤ n; j++)
            count ++;
    }
}
```

Have any Doubt ?

A $O(n \log n)$

B $O(n^2)$

C $O(n \log \log n)$

D $O(n^{3/2})$ Your answer is **Correct**

Solution :
(d)

$$\text{for } (i = 1; i \le n; i{+}{+}) \Rightarrow O(n)$$
$$\text{for } (j = 1; j \times j \le n; j{+}{+}) \Rightarrow O(\sqrt{n})$$
$$\text{count } {+}{+};$$
$$\text{Total time complexity} = O(n \times n^{1/2}) = O(n^{3/2})$$

QUESTION ANALYTICS +

Q. 10

Consider you are playing game of shooting balloon and you are expected to shoots 'n' balloon in the board. If you are a sharp shooter (100% accuracy) and for every 2 balloons you are able to shoot, one new balloon is inserted in the board then what is the time complexity of this shooting procedure if the board has to be emptied?

FAQ | Have any Doubt ?

A $O(1)$

B O(n) Correct Option

Solution :
(b)

Number of balloons originally	Total balloons shot
1	1
2	3
3	5
n	$2 \times n - 1$

C O($\log n$)

D O(n^2)

QUESTION ANALYTICS +

Q. 11

The minimum number of comparisons to find the biggest element in an unsorted list of size 15 is _____.

Have any Doubt ?

14 Your answer is **Correct**14

Solution :
(14)
(n – 1) comparisons are required to find the biggest element in an unsorted list of size n.

QUESTION ANALYTICS +

Q. 12

Consider the following program given below along with the input X[ ] and Y[ ]:

```
int Sequence(char *X, char *Y) {
    int m = strlen(X), n = strlen(Y);
    int l = LCS(X, Y, m, n);
    return (m + n – l);
}
int LCS( char *X, char *Y, int m, int n )
{
    int L[m + 1][n + 1];
    int i, j;
    for (i = 0; i ≤ m; i++) {
        for (j = 0; j ≤ n; j++) {
            if (i == 0 || j == 0)
                L[i][j] = 0;
            else if (X[i – 1] == Y[j – 1])
            L[i][j] = L[i – 1][j – 1] + 1;
            else
                L[i][j] = max(L[i – 1][j], L[i][j – 1]);
            }
        }
return L[m][n];
}
int main( ) {
    char X[ ] = "MADEEASY"; char Y[ ] = "EASYWAYS";
    printf("%d\n", Sequence(X, Y));
    return 0;
}
```

The value of printed by above program is _____.

Have any Doubt ?

12 Correct Option

Solution :
(12)
In the above program 1st we have to find the longest common subsequence which is 'EASY', the function Sequence( ) will gives the length of minimum length string which contain LCS of both the string i.e. length (MADEEASYWAYS) = 12.

QUESTION ANALYTICS +

Q. 13

Given a hash table with load factor of 300 that stores 10800 elements, the number of slots are _____.

Have any Doubt ?

36 Your answer is **Correct**36

Solution :
(36)

$$\text{Load factor} = \frac{\text{Number of elements}}{\text{Number of slots}}$$

$$300 = \frac{10800}{x}$$

$$x = 36$$

QUESTION ANALYTICS +

Q. 14

Consider the following scenario during insertion sort when the array looks like the following:
{25, 75, 95, 125, 80, 5, 10}

The number of comparisons that it will further take for the array to be completely sorted are _____.

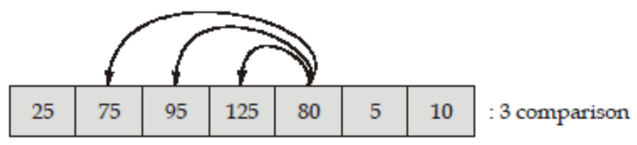
Have any Doubt ?

17 — Correct Option

Solution :
(17)

$$17 = 3 + 5 + 6 + 3$$

3 comparison, one each for 75, 95, 125



Your Answer is 20

QUESTION ANALYTICS

Q. 15

Consider two strings $x = "bcde"$ and $y = "cdgf"$, with following operations to convert x to y:
Insert a character into x (at any position).
Delete a character from x (from any position).
Replace a character in x by another character.
The minimum number of such operation required are _____.

Have any Doubt ?

3 — Correct Option

Solution :
13
1. Delete the beginning b from x. x becomes cde.
2. Replace character e in x by character g. x becomes cdg.
3. Insert character f at the end of x. x becomes $cdgf$.

QUESTION ANALYTICS

Q. 16

A professor gave Ramesh and Suresh task to compute nm. The algorithm used by Ramesh and Suresh are given below:

Ramesh

```
int power1(int x, int y)
{
    int temp;
    if (y = = 0) return 1;
    temp = power1 (x, y/2);
    if (y% 2 = = 0)
    return temp × temp;
    else
        return x × temp × temp;
}
int main ( )
{
    power1(a, b);
}
```

Suresh

```
int power2(int x, int y)
{
    if (y = = 0) return 1;
    else if (y% 2 = = 0)
    return power2 (x, y/2) × power2 (x, y/2)
    else
    return x × power2 (x, y/2) × power2 (x, y/2)
}
int main ( )
{
    power2(a, b);
}
```

The difference in time taken by Ramesh and Suresh algorithm to compute 44^{256} is _____.

Have any Doubt ?

248 — Correct Option

Solution :
(248)
Complexity of Ramesh algorithm = logm
Complexity of Suresh algorithm = m
$|8 - 256| = |248|$

QUESTION ANALYTICS

Q. 17

The preorder traversal of a Binary Search Tree is 100, 50, 20, 30, 60, 55, 150, 130, 140, 180, 170.
What would be its postorder traversal?

Have any Doubt ?

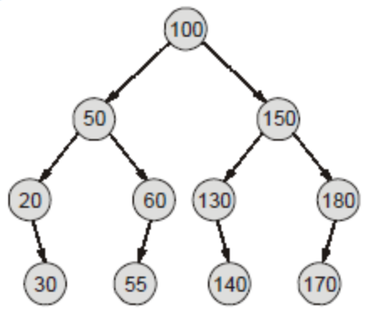
A 20, 30, 55, 60, 50, 130, 140, 180, 170, 150, 100

B 20, 30, 55, 60, 50, 140, 130, 180, 170, 150, 100

C 30, 20, 55, 60, 50, 140, 130, 170, 180, 150, 100 Your answer is **Correct**

Solution :

(c)
Inorder traversal of Binary Search Tree is in sorted order so,
Inorder traversal: 20, 30, 50, 55, 60, 100, 130, 140, 150, 170, 180
With inorder and preorder, unique Binary Search Tree can be determined.



Postorder traversal will be: 30, 20, 55, 60, 50, 140, 130, 170, 180, 150, 100

D 30, 20, 60, 55, 50, 130, 140, 180, 170, 150, 100

QUESTION ANALYTICS +

Q. 18

Assume an array of n integers and a function Max( ) given below:
Max( ) = {Find an element which is higher than its left and right element if any}
What is the worst case time complexity for Max( ) on an input array?

Have any Doubt ?

A $O(\log n)$

B $O(n)$ Correct Option

Solution :
(b)
Traverse the array element wise and for each element, compare it's value with it's left adjacent and right adjacent element (if it exists). If the condition is satisfied, print the elements.

C $O(n \log n)$

D $O(n^2)$

QUESTION ANALYTICS +

Q. 19

Given a degree sequence for 'n' vertices i.e., $d_1, d_2, \ldots d_v$. What is the worst case time complexity to determine if a simple graph is possible with the given degree sequence?

Have any Doubt ?

A $O(n)$

B $O(n^2 \log n)$ Correct Option

Solution :
(b)
Since a degree sequence is given, Havalle-Hakimi algorithm can be applied to it.
In this initially we have to sort the given degree sequence, and then decrement the degree sequence $a[i]^{th}$ time i starting from 0. The obtained output is sorted again.
In this way, the algorithm will take $O(n^2 \log n)$ time.

C $O(n \log n)$

D $O(n^2)$

QUESTION ANALYTICS +

Q. 20

What is the worst case time complexity to delete middle element from min heap of n distinct element?

Have any Doubt ?

A $O(\log n)$ Your answer is **Correct**

Solution :
(a)
We can find the middle element of the heap, by simply access is the $n/2^{th}$ element of the array which can be done in O(1) time. Further that element is replaced by the last element of the array in O(1) time. Now, perform the heapify operation which will take $O(\log n)$ time.

$$\text{Time taken} = O(1) + O(1) + O(\log n)$$
$$= O(\log n)$$

B $O(n)$

C $O(n \log n)$

D $O(n^2)$

QUESTION ANALYTICS +

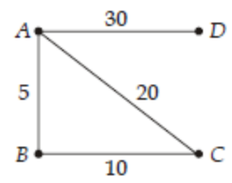
Q. 21

Consider the following statements:
S_1 : In a connected undirected graph $G = (V, E)$ with distinct edge costs, the cheapest edge belongs to every minimum spanning tree.
S_2 : In a connected undirected graph $G = (V, E)$ with distinct edge costs, the most expensive edge is excluded from every minimum spanning tree.
Which of the following is true?

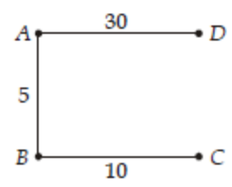
Have any Doubt ?

A Only S_1 — Your answer is **Correct**

Solution :
(a)
Since edges costs are distinct, so cheapest edge must be present in every minimum spanning tree while expensive edge is may not excluded from every minimum spanning tree.
Statement S_2 is false
Example:



MST will be:



So, most expensive edge is not excluded.

B Only S_2

C Both S_1 and S_2

D Neither S_1 nor S_2

QUESTION ANALYTICS +

Q. 22

Consider an array of numbers, where every number of array are same, is given as input. Which of the following statements is true, if we want to sort the array.
S_1 : Asymptotic running time of quick sort is $O(n\log n)$.
S_2 : Asymptotic running time of insertion sort is $O(n)$.
S_3 : Asymptotic running time of selection sort is $O(n)$.
S_4 : Asymptotic running time of merge sort is $O(n\log n)$.

Have any Doubt ?

A Only S_4

B Only S_2 and S_4 — Correct Option

Solution :
(b)
S_1 : Fully sorted array in case of quick sort becomes worst case.
Hence, time taken = $O(n^2)$.
S_2 : Insertion sort on almost sorted array takes $O(n)$ time.
S_3 : Selection sort always takes $O(n^2)$ time.
S_4 : Merge sort always takes $O(n\log n)$ time.

C Only S_1, S_2 and S_4

D Only S_3 and S_4 — Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 23

In an adjacency list representation of an undirected graph $G = (v, e)$, for any 2 sets of vertices v_1 and v_2 let, distance (v_1, v_2) be defined as the minimum of the length of shortest distance between a vertex in v_1 and v_2, if $v_1 \cap v_2 \neq \phi$, then distance $(v_1, v_2) = 0$. The most optimal time complexity for computing distance (v_1, v_2) is:

Have any Doubt ?

A $O(|v| + |e|)$ — Correct Option

Solution :
(a)
Take 2 new vertices a and b. Connect a to all vertices in v_1 and b to all vertices in v_2.
Perform BFS from a to b. The length of the path obtained minus 2 (one edge from a and one edge to b) will give the required result.

B $O(|v| \times |e|)$ — Your answer is **Wrong**

C $O(|v| \times |e| + |e| \times |v|)$

D $O(|e| \times |v^2|)$

QUESTION ANALYTICS +

QUESTION ANALYTICS +

Q. 24

Match **List-I (Algorithm)** with **List-II (Complexity)** and select the correct answer using the codes given below the lists:

List-I (Algorithm)	List-II (Complexity)
A. Merge sort with no extra space.	1. $O(n\log n)$
B. Finding number of inversion pair.	2. $O(\log n)$
C. Topological sort with 'n' edges and n vertices.	3. $O(n^2)$
D. Finding median of two sorted array each of size 'n'.	4. $O(n)$

Codes:

	A	B	C	D
(a)	2	3	1	4
(b)	3	1	4	2
(c)	2	3	4	1
(d)	1	3	4	2

Have any Doubt ?

A a

B b — Correct Option

Solution :
(b)

A. Merge sort is outplace algorithm, if extra space is not allowed the it take $O(n^2)$ time.
B. Finding number of inversion pair using merge sort takes $O(n\log n)$ time.
C. Topological sort take $O(n)$ time using DFS.
D. Finding median of two sorted array takes $O(\log n)$ time.

C c

D d — Your answer is **Wrong**

QUESTION ANALYTICS +

Q. 25

In a directed graph G (number of vertices = n), super-sink is a vertex whose indegree is $(n-1)$ and outdegree is zero. What will be the optimal time complexity to determine if such a vertex exists in the graph or not if adjacency matrix representation of graph is used?

Have any Doubt ?

A $O(n)$ — Correct Option

Solution :
(a)

A vertex is super sink if and only if $M[i, j] = 0\ \forall\ j$ and $M[j, i] = 1\ \forall\ j \neq i$.
Traverse over the matrix and check the above condition,
Using below program we can find super-sink in directed graph:

```
i = 0;
do
{
      j = i + 1;
      while ((j < n) && !A[i][j]) j++;
      if (j < n) i = j;
} while (i < n);
flag = 1;
for (j = 0; j < n; j++)
      if ((j! = i) && (A[i][j] || !A[j][i])) flag = 0;
          if (flag) printf("Sink exists");
          else printf ("Sink does not exist");
```

B $O(\log n)$

C $O(n^2)$

D $O(\log\log n)$

QUESTION ANALYTICS +

Q. 26

Consider the partial pseudocode given below used to calculate median of combination of 2 sorted arrays (arr1 and arr2) of equal size n.

```
int MEDIAN(int arr1[ ], int arr2[ ], int n) {
m1 = median(arr1)   / * calculates median of array elements, If n is even, median = (middle
element + next element)/2, if n is odd, median = middle element */
m2 = median(arr2)
if (m1 == m2) return m1;
if (m1 < m2) {
    if (no of elements == even) return S1;
    return MEDIAN(arr1 + n/2, arr2, n – n/2);
}
if (m1 < m2) {
        if (no of elements == even)
            return MEDIAN(arr2 + n/2 – 1, arr1, n – n/2 + 1);
        return S2;
```

}
}

The missing argument lists are respectively:

(a) MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1); MEDIAN(arr2 + n/2, arr1, n – n/2)

(b) MEDIAN(arr2 + n/2, arr1, n – n/2); MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1)

(c) MEDIAN(arr2 + n/2, arr1, n – n/2) ; MEDIAN(arr2 + n/2, arr1, n – n/2)

(d) MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1); MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1)

Have any Doubt ?

A MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1); MEDIAN(arr2 + n/2, arr1, n – n/2) — Correct Option

Solution :
(a)
If m1 is greater than m2, then median is present in between first element of arr1 to m1 or from m2 to last element of arr2. If m2 is greater than m1, then median is present in between m1 to last element of arr1 or from first element of arr2 to m2.

B MEDIAN(arr2 + n/2, arr1, n – n/2); MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1)

C MEDIAN(arr2 + n/2, arr1, n – n/2) ; MEDIAN(arr2 + n/2, arr1, n – n/2)

D MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1); MEDIAN(arr1 + n/2 – 1, arr2, n – n/2 + 1)

QUESTION ANALYTICS +

Q. 27

Consider $G = (V, E)$ be a connected graph with n vertices and m edges with distinct positive edge weight. $T = (V, E')$ be a spanning tree of G and bottleneck edge of T is the edge with greatest cost in T. A spanning tree T of G is a minimum bottleneck spanning tree if there is no spanning tree T′ of G with cheaper bottleneck edge. Which of the following statement true?

S_1 : Every minimum bottleneck tree of G is a minimum spanning tree of G.

S_2 : Every minimum spanning tree of G is a minimum bottleneck tree of G.

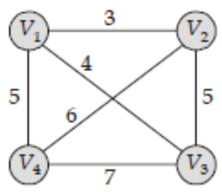
Have any Doubt ?

A Only S_1

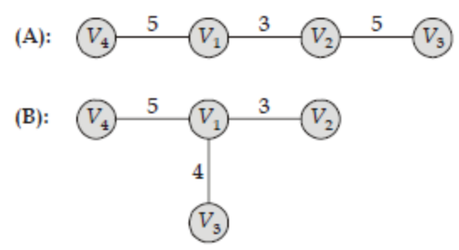
B Only S_2 — Correct Option

Solution :
(b)

S_1 : Consider the following graph:

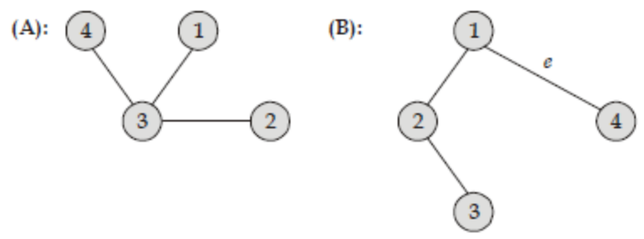


Minimum bottleneck spanning tree:



However only (b) is MST therefore S_1 is false.

S_2 : Consider 2 MST's:



Where A has a lighter bottleneck edge. This means that B has an edge 'e' that is heavier than every edge of A. If we include this edge in A, it will form a cycle in which e would be heaviest. This contradicts the definition of MST.

Thus, 'e' cant be present in B, meaning that both A and B have same bottleneck edge.

Thus, S_2 is true.

C Both S_1 and S_2

D Neither S_1 nor S_2

QUESTION ANALYTICS +

Q. 28

Given two sequences A and B. Let $X(A, B)$ denote the number of times that A appears as subsequence of B i.e. sequence ab appears 4 times as a subsequence of aebabdb. Let A_i denotes the first i characters of string A and $A[i]$ denote the $_i$th character. Which of the following will computes the recurrence relation $C(A_i, B_j)$?

$$C(A_i, B_j) = \begin{cases} 1; & \text{if } (i = 0) \\ 0; & \text{if } (i > 0 \text{ and } j = 0) \\ \text{----} & \text{----------} \\ \text{----} & \text{----------} \end{cases}$$

Have any Doubt ?

A $C(A_i, B_{j-1})$; if $A[i] \neq B[j]$
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] = B[j]$ — Correct Option

Solution :
(a)

$$C(A_i, B_j) = \begin{cases} 1; & \text{if } (i = 0) \\ 0; & \text{if } (i > 0 \text{ and } j = 0) \\ C(A_i, B_{j-1}); & \text{if } (A[i] \neq B[i]) \text{ when elements are not matched.} \\ C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1}); & \text{if } (A[i] = B[i]) \text{ when elements are matched.} \end{cases}$$

B $C(A_{i-1}, B_j)$; if $A[i] = B[j]$
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] = B[j]$

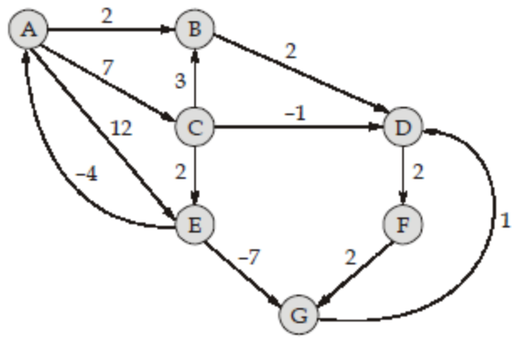
C $C(A_i, B_{j-1})$; if $A[i] = B[j]$ — Your answer is **Wrong**
$C(A_i, B_{j-1}) + C(A_{i-1}, B_{j-1})$; if $A[i] \neq B[j]$

D None of the above

QUESTION ANALYTICS +

Q. 29

While applying Dijkstra algorithm on given graph, it is already known that it is computing wrong path to some of the vertices. The number of wrong path computed are _____.



Have any Doubt ?

3 — Correct Option

Solution :
3

	Using Dijkstra algorithm		Actual Shortest path	
A → B	2	AB	2	AB
C	7	AC	7	AC
D	4	ABD	3	ACEGD
E	9	ACE	9	ACE
F	6	ABDF	5	ACEGDF
G	8	ABDFG	2	ACEG

So, there are 3 wrong path calculated to D, F and G vertices.

QUESTION ANALYTICS +

Q. 30

Suppose to encode a text with the characters a, b, c, d, e, f with the following frequencies:

Letters	a	b	c	d	e	f
Frequency	15	17	25	23	18	12

If followed fixed length code each character takes 1 bits. If compression technique used is Huffman coding, then the average length of the encoded message in bits is _____. (Upto 2 decimal places)

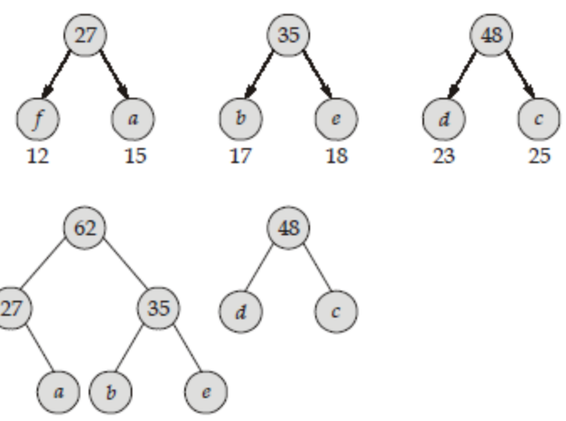
Have any Doubt ?

2.56 [2.55 - 2.57] — Correct Option

Solution :
2.56 [2.55 - 2.57]

Total characters are 110 bits.

As Huffman tree:



According to Huffman encoding
$a = 001 \Rightarrow 3$ bit
$b = 010 \Rightarrow 3$ bit
$c = 11 \Rightarrow 2$ bit
$d = 10 \Rightarrow 2$ bit
$e = 011 \Rightarrow 3$ bit
$f = 000 \Rightarrow 3$ bit

$$\text{Number of bits needed} = (3 \times 12) + (3 \times 15) + (3 \times 17) + (3 \times 18) + (2 \times 23) + (2 \times 25) = 282$$

$$\text{Average bits} = \frac{282}{110} = 2.56 \text{ bits/char}$$

QUESTION ANALYTICS +

Q. 31

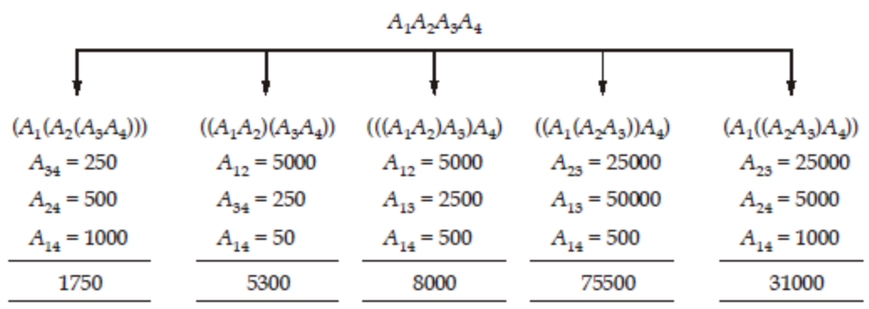
Consider a matrix multiplication chain A_1, A_2, A_3 *and* A_4, where matrices A_1, A_2, A_3 and A_4 are of dimensions 10 × 100, 100 × 5, 5 × 50 and 50 × 1 respectively.
The difference between the minimum and maximum number of scalar multiplications required to find the product of A_1, A_2, A_3 *and* A_4 using basic matrix multiplication method is _____.

Have any Doubt ?

73750

73750
Your answer is **Correct**

Solution :
73750

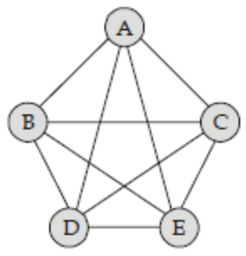


So, difference = |Maximum multiplication – Minimum multiplication|
= |75500 – 1750|
= 73750

QUESTION ANALYTICS

Q. 32

Consider a graph '*G*' given below:



If edges are having weight 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. Then maximum possible weight that a minimum weight spanning tree of *G* can have is _____.

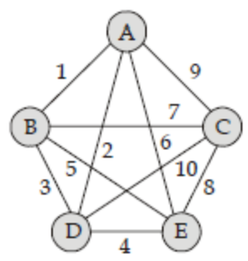
Have any Doubt ?

14

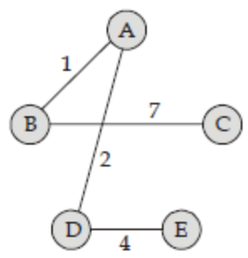
Correct Option

Solution :
14

Consider edges weights are numbered as below:



So minimum cost spanning tree will be:



In worst case, cost of MST will be 1 + 2 + 4 + 7 = 14.

Your Answer is 15

QUESTION ANALYTICS

Q. 33

A multistage graph is a directed graph in which the nodes can be divided into a set of stages such that all edges are from a stage to next stage only. There is no edge between vertices of same stage and from a vertex of current stage to previous stage. The recurrence relation for finding the shortest path distance between two vertices (i, j) is given below:

$$\text{MSG}[i] = \min_{\forall j > i} \begin{Bmatrix} 0 & \text{if } j \in F \\ e(i,j) + \text{MSG}[j] & \text{if } j \notin F \end{Bmatrix}$$

where $e(i, j)$ = weight of edge connecting i and j. F is the final stage.
The shortest path value between source and destination in the following graph is _____.

Have any Doubt ?

16

Correct Option

Solution :
16

MSG (12) = 0 since $12 \in F$
MSG (11) = $e(11, 12)$ + MSG (12) = 5 + 0 = 5

$$\begin{aligned} \text{MSG (10)} &= 2 \\ \text{MSG (9)} &= 4 \\ \text{MSG (8)} &= \min(e(8, 10) + \text{MSG (10)}, e(8, 11) + \text{MSG (11)}) \\ &= \min(5 + 2, 6 + 5) = 7 \end{aligned}$$

and so on .....

Finally,

1	2	3	4	5	6	7	8	9	10	11	12
16	7	9	18	15	7	5	7	4	2	5	0

So, the shortest path value between source and destination is 16.

[Trick: If the graph given is multistage graph, than greedy approach can be used from backwards for choosing the minimum weight, edge until reach source.]

QUESTION ANALYTICS +