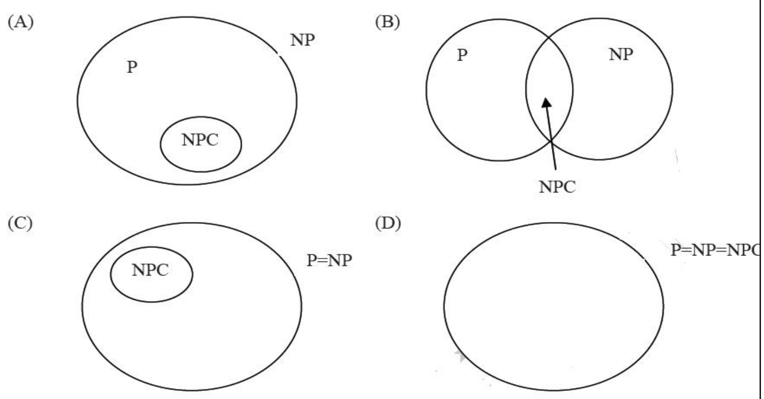
AUTUMN END SEMESTER EXAMINATION-2022
DESIGN & ANALYSIS OF ALGORITHMS
[CS 2012]
SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY, DEEMED TO BE UNIVERSITY

Time: 3 Hours **Full Marks: 50**

Answer any SIX questions.
Question paper consists of four SECTIONS i.e. A, B, C and D.
Section A is compulsory.
Attempt minimum one question each from Sections B, C, D.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

Solution and Evaluation Scheme

1.		Answer the following questions.	$[1 \times 10]$
	(a)	Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?	
	Solution	**2 times** **Correct Answer: 1 Mark** **Wrong Answer: 0 Marks**	
	(b)	Find the worst case time complexity of quick sort and its recurrence. (a) Time complexity is O(n^2) and recurrence is T(n) = T(n-2) + O(n) (b) Time complexity is O(n^2) and recurrence is T(n) = T(n-1) + O(n) (c) Time complexity is O(nLogn) and recurrence is T(n) = 2T(n/2) Time complexity is O(nLogn) and recurrence is T(n) = T(n/10) + T(9n/10) + O(n)	
	Solution	**Correct Choice: A and B are correct.** **Correct Answer: 1 Mark** **Wrong Answer: 0 Marks**	
	(c)	Solve the following recurrence relation? T(n) = 7T(n/2) + 3n^2 + 2	
	Solution	**a = 7, b = 2, and f(n) = 3n^2 + 2** **So, f(n) = O(n^c), where c = 2.** **logb(a) = log2(7) = 2.81 > 2** **It follows from the first case of the master theorem that** **T(n) = θ(n^2.8) and implies O(n^2.8) as well as O(n^3)** **Correct Answer: 1 Marks** **Wrong Answer: 0 Marks**	
	(d)	Sort the following functions in the decreasing order of their asymptotic (big-O) complexity: f1(n) = n^√n , f2(n) = 2^n, f3(n) = (1.000001)^n , f4(n) = n^(10)*2^(n/2)	
	Solution	**f2> f4> f3> f1** **Correct Answer: 1 Mark** **Wrong Answer: 0 Marks**	

	(e)	Suppose a polynomial time algorithm is discovered that correctly computes the largest clique in a given graph. In this scenario, which one of the following represents the correct Venn diagram of the complexity classes P, NP and NP Complete (NPC)?  	
	Solution	**Choice:D** **Correct Choice: 1 Mark** **Wrong Answer: 0 Marks**	
	(f)	Compute the minimum number of scalar multiplications required to multiply four matrices having dimensions 20 x 15, 15 x 30, 30 x 5 and 5 x 40 (a) 6050 (b) 7500 (c) 7750 (d) 12000	
	Solution	**Choice:C** **Correct Choice: 1 Mark** **Wrong Answer: 0 Marks**	
	(g)	Let us consider, two sequences "QPQRR" and "PQPRQRP". Determine the LCS of these sequences. (a) QPRR (b) PQRR (c) QPQR (d) All of the above	
	Solution	**Choice:D** **Correct Choice: 1 Mark** **Wrong Answer: 0 Marks**	
	(h)	Let us consider two problem A and B. The problem B is NP complete. The problem A reduces to problem B in polynomial time. Determine which of the following statement is correct? (a) If A can be solved in polynomial time then B can also be solved in polynomial time (b) A is NP complete problem (c) A is NP hard problem (d) A is in NP but not in NP complete	
	Solution	**Choice: None of the above** **Correct Choice: 1 Mark** **Wrong Answer: 0 Marks**	
	(i)	Let us consider a file consists of six characters such as A, B, C, D, E, and F having probabilities of 1/2, 1/4, 1/8, 1/16, 1/32, and 1/32 respectively. Determine which of the following codes (huffman) for the letters A, B, C, D, E, and F? (a) 0, 10, 110, 1110, 11110, 11111 (b) 11, 10, 01, 001, 0001, 0000 (c) 11, 10, 011, 010, 001, 000 (d) 110, 100, 010, 000, 001, 111	
	Solution	**Choice:A** **Correct Choice: 1 Mark** **Wrong Answer: 0 Marks**	

(j) Determine which of the following algorithm is based on the principle of dynamic programming.
(a) Floyd Warshll Algorithm for all pairs shortest path
(b) Dijkstra Algorithm for single source shortest paths
(c) Fractional Knapsack problem
(d) Prim's Minimum Spanning Tree

Solution

Choice:A
Correct Choice: 1 Mark
Wrong Answer: 0 Marks

SECTION-B

2. (a) Find an optimal number of Scalar multiplication required of a MATRIX-CHAIN product whose sequence of dimensions are < 5,10,3,12,5,50,6>. Write the recursive procedure of matrix chain multiplication and its time complexity. [4]

Solution

Answer: The m-table and s-table are given as follows.

m-table

i \ j	6	5	4	3	2	1
6	0					
5	1500	0				
4	1860	3000	0			
3	1770	930	180	0		
2	1950	2430	330	360	0	
1	2010	1655	405	330	150	0

s-table

i \ j	6	5	4	3	2
5	5				
4	4	4			
3	4	4	3		
2	2	2	2	2	
1	2	4	2	2	1

According to s-table shown above, the optimal parenthesization is $(A_1A_2)((A_3A_4)(A_5A_6))$.

```
MATRIX-CHAIN(i, j)
    IF i = j THEN return 0
    m = ∞
    FOR k = i TO j − 1 DO
        q = MATRIX-CHAIN(i, k) + MATRIX-CHAIN(k + 1, j) + p_{i−1} · p_k · p_j
        IF q < m THEN m = q
    OD
    Return m
END MATRIX-CHAIN

Return MATRIX-CHAIN(1, n)
```

- Running time:

$$
\begin{aligned}
T(n) &= \sum_{k=1}^{n-1}(T(k) + T(n-k) + O(1)) \\
&= 2 \cdot \sum_{k=1}^{n-1} T(k) + O(n) \\
&\geq 2 \cdot T(n-1) \\
&\geq 2 \cdot 2 \cdot T(n-2) \\
&\geq 2 \cdot 2 \cdot 2 \ldots \\
&= 2^n
\end{aligned}
$$

Scheme:

Finding Minimum Number of Scalar Multiplications for given matrices: 2 mark

For writing MCM recursive algorithm and time complexity: 2 Mark

Partially correct : 2 marks can be awarded

Wrong Answer: 0 Marks

	(b)	Use a dynamic programming algorithm to find the Longest Common Subsequence between the following two sequences: X = ababaabaa and Y = aababaab. Write the recursive procedure of LCS and its time complexity.	[4]

Solution

Given X = “ababaabaa”, Y = “aababaab”

X | Y	y_i	a	a	b	a	b	a	a	b
x_i	0	0	0	0	0	0	0	0	0
a	0	↖ 1	↖ 1	← 1	↖ 1	← 1	↖ 1	↖ 1	← 1
b	0	↑ 1	↑ 1	↖ 2	← 2	↖ 2	← 2	← 2	↖ 2
a	0	↖ 1	↖ 2	← 2	↖ 3	← 2	↖ 3	↖ 3	← 3
b	0	↑ 1	↑ 2	↖ 3	← 3	↖ 4	← 4	← 4	↖ 4
a	0	↖ 1	↖ 2	↑ 3	↖ 4	← 4	↖ 5	↖ 5	← 5
a	0	↖ 1	↖ 2	↑ 3	↖ 4	← 4	↖ 5	↖ 6	← 6
b	0	↑ 1	↑ 2	↖ 3	↑ 4	↖ 5	← 5	↑ 6	↖ 7
a	0	↖ 1	↖ 2	↑ 3	↖ 4	↑ 5	↖ 6	↖ 6	↑ 7
a	0	↖ 1	↖ 2	↑ 3	↖ 4	↑ 5	↖ 6	↖ 7	← 7

The Longest Common Subsequence is: “aababaa”; Length = 7

Recursive algorithm for LCS and time complexity

LCS-LENGTH(X, Y, m, n)

```
for i ← 1 to m
    do c[i, 0] ← 0
for j ← 0 to n
    do c[0, j] ← 0
for i ← 1 to m
    do for j ← 1 to n
        do if x_i = y_j
            then c[i, j] ← c[i - 1, j - 1] + 1
                 b[i, j ] ← "↖"
            else if c[i - 1, j] ≥ c[i, j - 1]
                then c[i, j] ← c[i - 1, j]
                     b[i, j] ← "↑"
                else c[i, j] ← c[i, j - 1]
                     b[i, j] ← "←"
return c and b
```

The length of the LCS if one of the sequences is empty i.e. its length is zero

Case 1: $x_i = y_j$

Case 2: $x_i \neq y_j$

Running time: $\Theta(mn)$

PRINT-LCS(b, X, i, j)

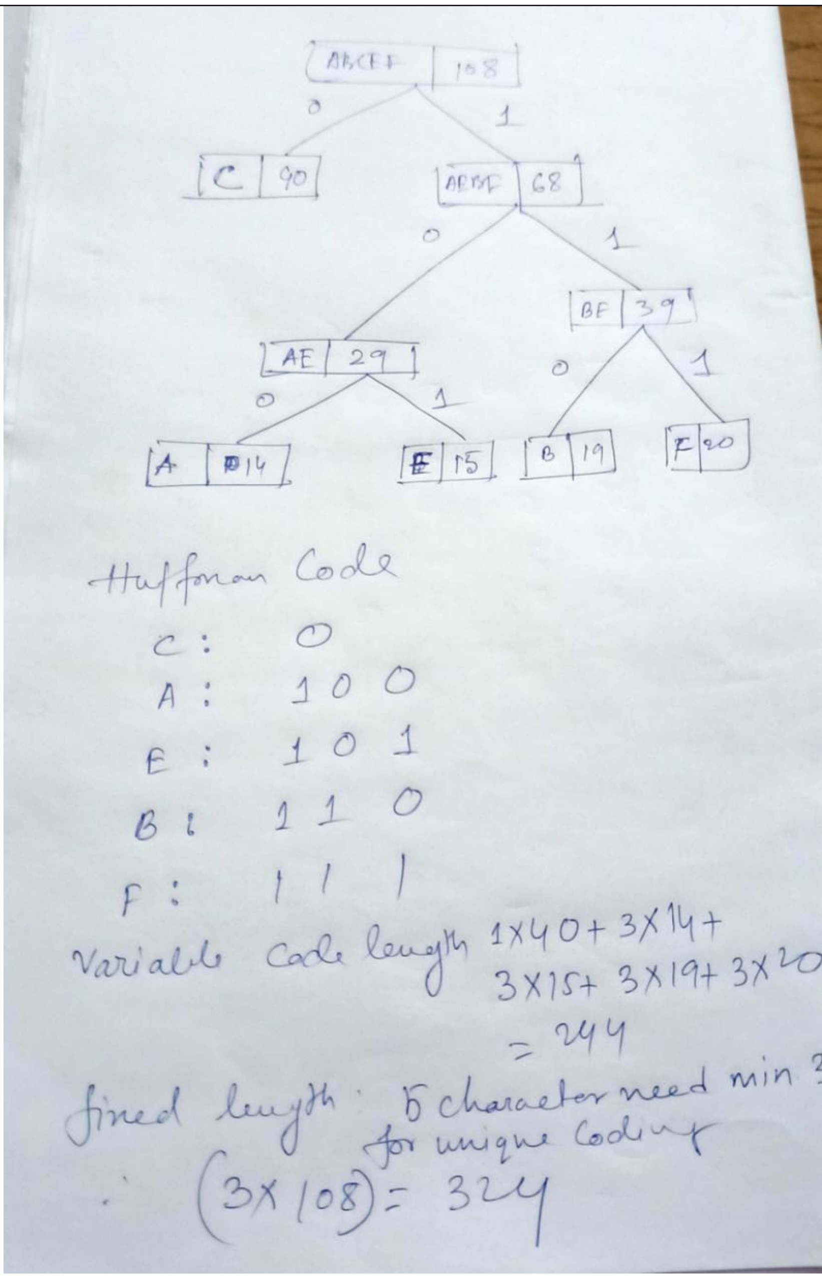
```
if i = 0 or j = 0
  then return
if b[i, j] = " ↖"
   then PRINT-LCS(b, X, i - 1, j - 1)
        print x_i
elseif b[i, j] = "↑"
        then PRINT-LCS(b, X, i - 1, j)
        else PRINT-LCS(b, X, i, j - 1)
```

Running time: $\Theta(m + n)$

Initial call: PRINT-LCS(b, X, length[X], length[Y])

Scheme:

Finding LCS for given strings using dynamic Programming Approach: 2 mark

		For writing LCS algorithm and time complexity: 2 Mark **Partially correct answer: 2 marks** **Wrong Answer: 0 Marks**	
3.	(a)	What will be optimal Huffman code for the following set of symbol having given frequencies: A:14, B:19, C:40, E: 15, f,20. Draw the decode tree for both fixed and variable length encoding scheme for the above data. Explain which method compress more amount of data.	[4]
	Solution	 **Scheme:** **For drawing decode tree for both fixed and variable length(huffman coding) encoding scheme 3 mark** **Finding and comparing both techniques length 1 mark** **Partially correct answer: 2 marks** **Wrong answer: 0 Mark**	
	(b)	Find an optimal solution to the knapsack instance n=7, W=15. (v1, v2, v3, v4, v5, v6, v7) = (5, 15, 10, 7, 6, 20, 3) and (w1, w2, w3, w4, w5, w6, w7) = (2, 3, 5, 6, 1, 4, 1), where n is the number of items, W is the knapsack capacity that thief can carry, v_i stands for value or profit w_i stands for	[4]

weight of the i^{th} element.

Solution

N = 7 and W = 15

(v1, v2, v3, v4, v5, v6, v7) = (5, 15, 10, 7, 6, 20, 3) and

(w1, w2, w3, w4, w5, w6, w7) = (2, 3, 5, 6, 1, 4, 1),

Item Number	1	2	3	4	5	6	7
vi	5	15	10	7	6	20	3
wi	2	3	5	6	1	4	1
Vi/wi	2.5	5	2	1.16	6	5	3

Item put in the knapsack

	W	Profit
Initially	15	0
After inserting item 5	15-1=14	6
After inserting item 2	14-3=11	6+15 = 21
After inserting item 6	11-4 = 7	21+20=41
After inserting item 7	7-1=6	41+3 = 44
After inserting item 1	6-2 = 4	44+5 = 49
After inserting 4/5 units of item 3	4-4=0	49+(10*(4/5))=57

The optimal profit is 57.

Scheme:

Finding optimal solution for the given data using knapsack technique: 4 marks

Partially correct answer:2 marks

Wrong answer:0 Marks

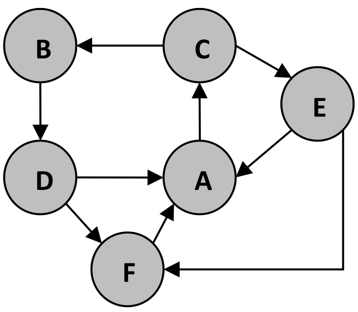
SECTION-C

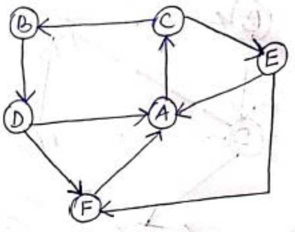
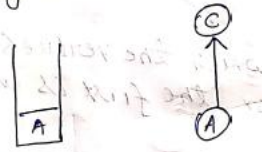
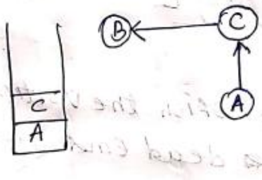
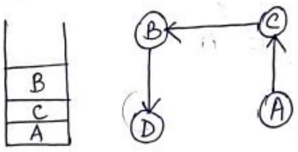
4. (a) Consider the following graph and solve the followings [4]

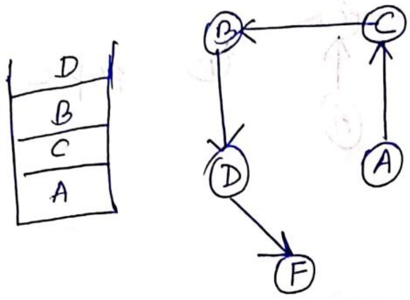
a) Compute the DFS tree and draw the tree edges, forward edges, back edges and cross edges.

b) Write the order in which the vertices were reached for the first (i.e.pushed into the stack)

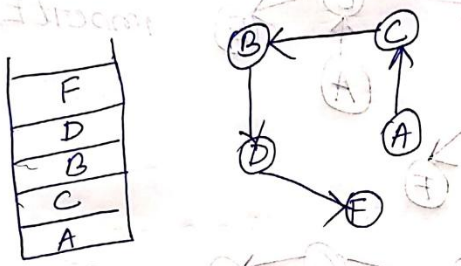
c) Write the order in which the vertices became dead ends (i.e. popped from the stack)



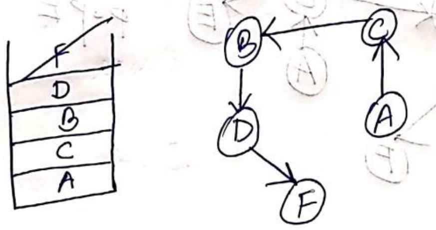
	Solution	4 a)  Computation of DFS tree  insert A  insert C  insert B	



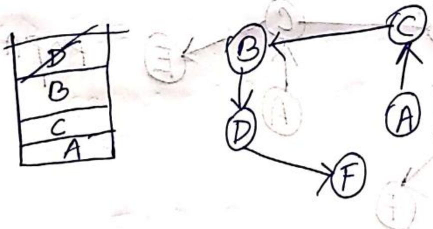
insert D



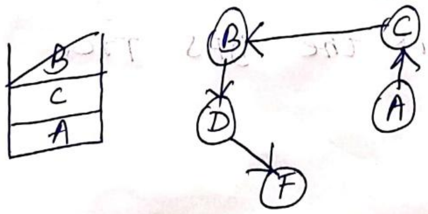
insert F



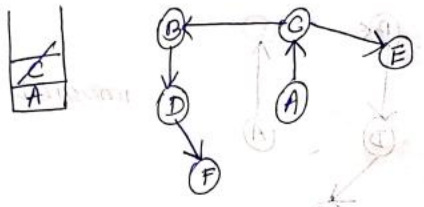
pop F



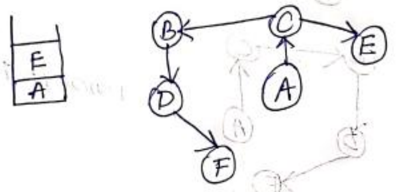
pop D



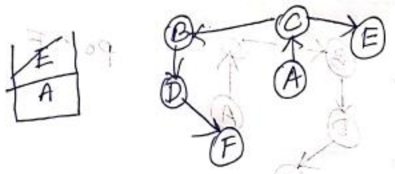
pop B



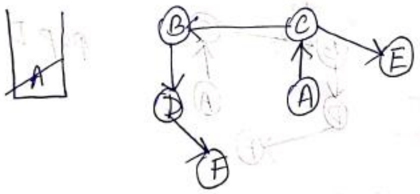
pop C



insert E



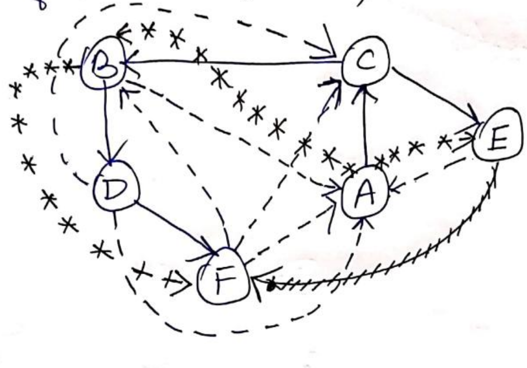
pop E



pop A

The above is the DFS Tree

Therefore the DFS Tree ~~is~~ with tree edge, forward, back, cross edge is-



- - - -
Back edge

———
Tree edge.

/////////
Cross edge.

* * * *
Forward edge.

b) The order in which the vertices were reached for the first is as follows

A, C, B, D, F, E

c) The order in which the vertices become a dead end is as follows.

F, D, B, C, E, A

Scheme:

A)For computing and drawing the DFS tree edges, forward edges, back edges and cross edges --2 marks

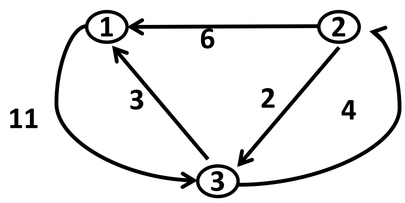
B) For writing the order in which the vertices were reached for the first (i.e.pushed into the stack)--1 marks

C) For writing the order in which the vertices became dead ends (i.e. popped from the stack)---1 mark

Partially correct answers: 2 Marks

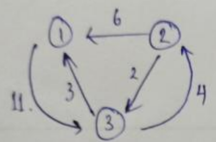
Wrong answers: 0 Marks

(b) Find all pair shortest path using Floyd Warshall algorithm for the following graph. [4]



Solution

4. b).



$$A^0 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & \infty & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 4 & 0 \end{array}$$

K=1

$$A^1 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & \infty & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 4 & 0 \end{array}$$

k=2

$$A^2 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & \infty & 11 \\ 2 & 6 & 0 & 2 \\ 3 & 3 & 4 & 0 \end{array}$$

K=3

$$A^3 = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 15 & 11 \\ 2 & 5 & 0 & 2 \\ 3 & 3 & 4 & 0 \end{array}$$

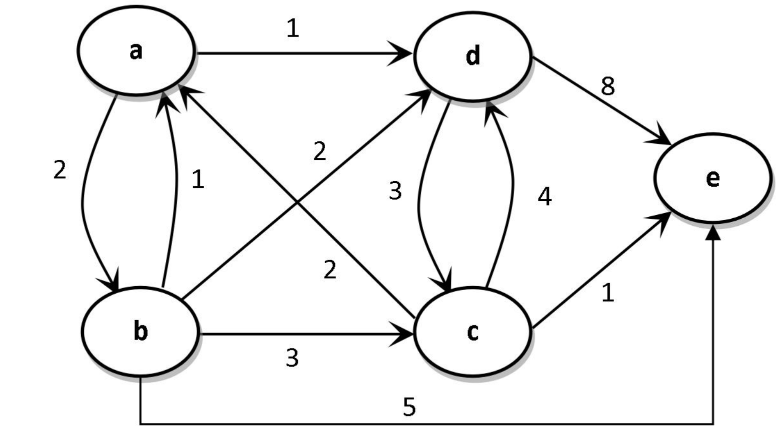
Scheme:

For finding all pair shortest path using Floyd Warshall algorithm --4 Marks

Partially Correct answer--2 marks

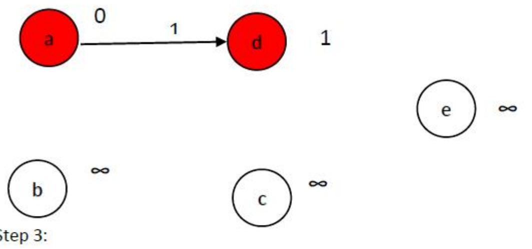
Wrong answer--0 mark

5. (a) Use suitable shortest path algorithm to find out shortest path from vertex 'a' to all other vertices. Show all the steps. [4]



Solution

Here, 1 is the minimum. So, vertex 'd' mark as visited vertex, and updated distance 'a' to 'd' is dist(a) dist(a,d)= 0+1=1



Step 3:

Queue / Visited Vertex	a	b	c	d	e
	0	∞	∞	∞	∞
a		0+2=2	∞	0+1=1	∞
d		0+2=2	1+3=4		1+8=9

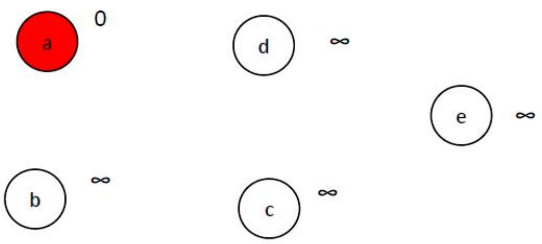
Here, 2 is the minimum. So, vertex 'b' mark as visited vertex, and the updated distance 'a' to 'b' is dist(a) dist(a,b)= 0+2=2

There is no negative weight cycle in the given graph. So, we can use **Dijkstra Algorithm** to find the shortest path from vertex 'a' to other vertices.

Step 1:

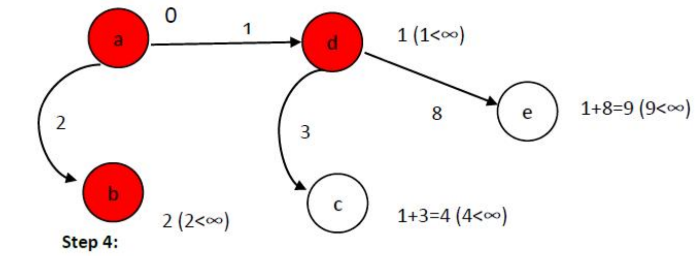
Queue / Visited Vertex	a	b	c	d	e
	0	∞	∞	∞	∞

Here, distance(a)=0, which is also 0 minimum. So, vertex 'a' mark as visited vertex.



Step 2:

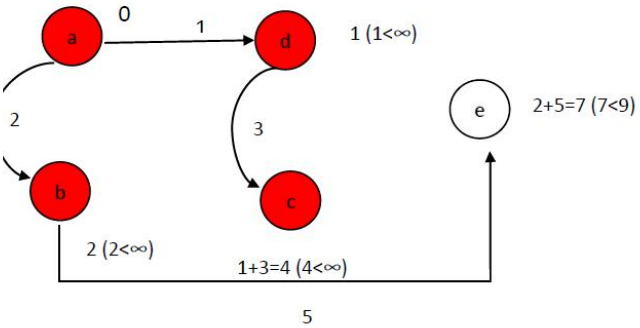
Queue / Visited Vertex	a	b	c	d	e
	0	∞	∞	∞	∞
A		2	∞	1	∞



Step 4:

Visited Vertex \ Queue	a	b	c	d	E
	0	∞	∞	∞	∞
a		0+2=2	∞	0+1=1	∞
d		0+2=2	1+3=4		1+8=9
b			4		2+5=7

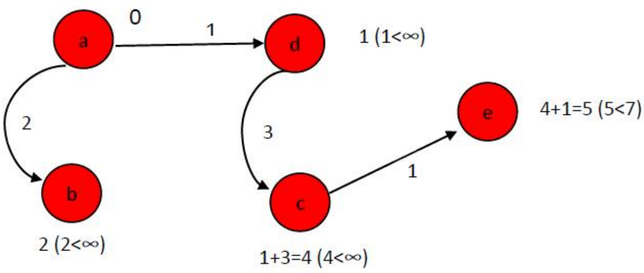
Here, via ‘b’, the distance a to c is dist (b)+dist(b,c) = 2+3=5, which is greater than previous distance 4. So, visit ‘c’ via vertex ‘d’. Also, we can visit now vertex ‘e’ via vertex ‘b’ with distance with distance dist(b)+dist(b,e) = 2+5=7<9. Hence, update the distance of vertex ‘e’. Now, minimum distance of vertex ‘c’ is 4, so next visited vertex is ‘c’.



Step 5:

Visited Vertex \ Queue	a	b	c	d	e
	0	∞	∞	∞	∞
A		0+2=2	∞	0+1=1	∞
D		0+2=2	1+3=4		1+8=9
B			4		2+5=7
C					4+1=5
e					

Now we can visit now vertex ‘e’ via vertex ‘c’ with distance with distance dist(c)+dist(c,e) = 4+1=5<7. Hence, update the distance of vertex ‘e’. Now, minimum distance is 5, so next visited vertex is ‘e’.



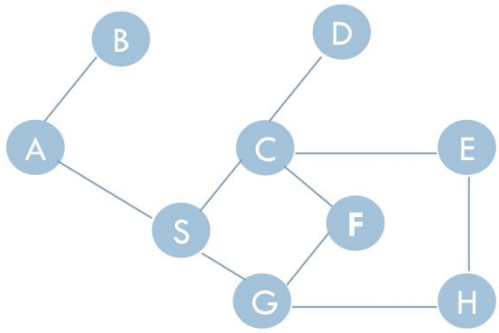
Scheme:

To find shortest path from vertex ‘a’ to all other vertices with all the steps(either disjtras or bellman ford) --4 mark

Partially correct answer: 2 marks

Wrong answer: 0 marks

(b) Consider the following graph and compute the BFS tree. [4]



BFS Algorithm:

As in question source vertex is not given, students can take any vertex as source vertex.

Sample solution with ‘A’ as source vertex.

Step-1:

u={A}, w={B,S}

~~vertex~~	Deleted from Queue

vertex	Inserted into Queue

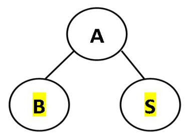
VISITED

A	B	C	D	E	F	G	H	S
1	1	0	0	0	0	0	0	1

Queue

~~A~~	B	S	

Tree:



Step-2:

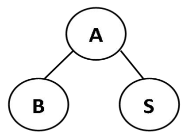
u={B}, w={A}

VISITED

A	B	C	D	E	F	G	H	S
1	1	0	0	0	0	0	0	1

Queue

~~A~~	~~B~~	S	



Step-3:

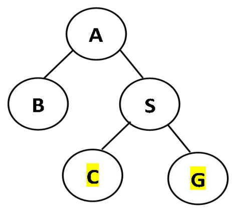
u={S}, w={A,C,G}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	0	0	0	1	0	1

Queue

~~A~~	~~B~~	~~S~~	C	G	



Step-4:

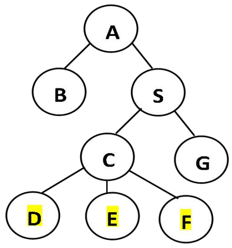
u={C}, w={S,D,E,F}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	0	1

Queue

A	B	S	C	G	D	E	F



Step-5:

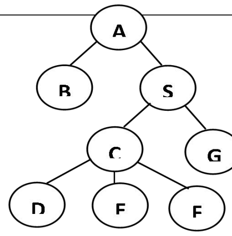
u={G}, w={F,S}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	0	1

Queue

A	B	S	C	G	D	E	F



Step-6:

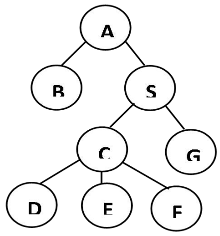
u={D}, w={C}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	0	1

Queue

A	B	S	C	G	D	E	F



Step-7:

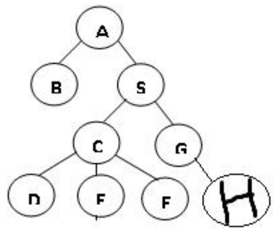
u={E}, w={C,H}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	1	1

Queue

A	B	S	C	G	D	E	F	H



Step-8:

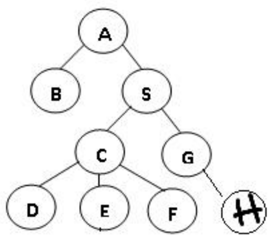
u={F}, w={C,G}

VISITED

A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	1	1

Queue

A	B	S	C	G	D	E	F	H



Step-9:

u={H}, w={E,G}

VISITED

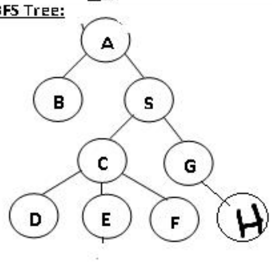
A	B	C	D	E	F	G	H	S
1	1	1	1	1	1	1	1	1

Queue

A	B	S	C	G	D	E	F	H

Final BFS Traversal:
A, B, S, C, G, D, E, F ,H

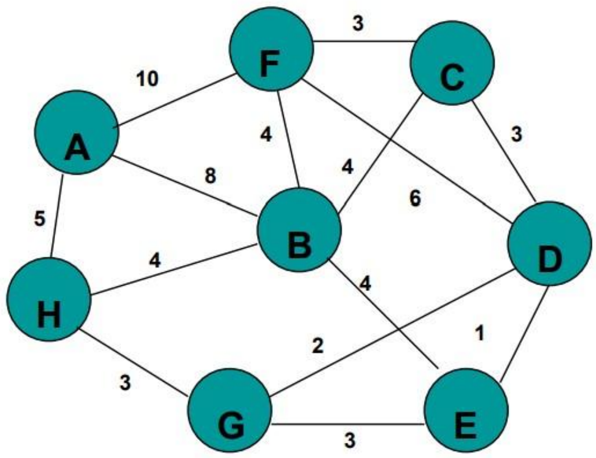
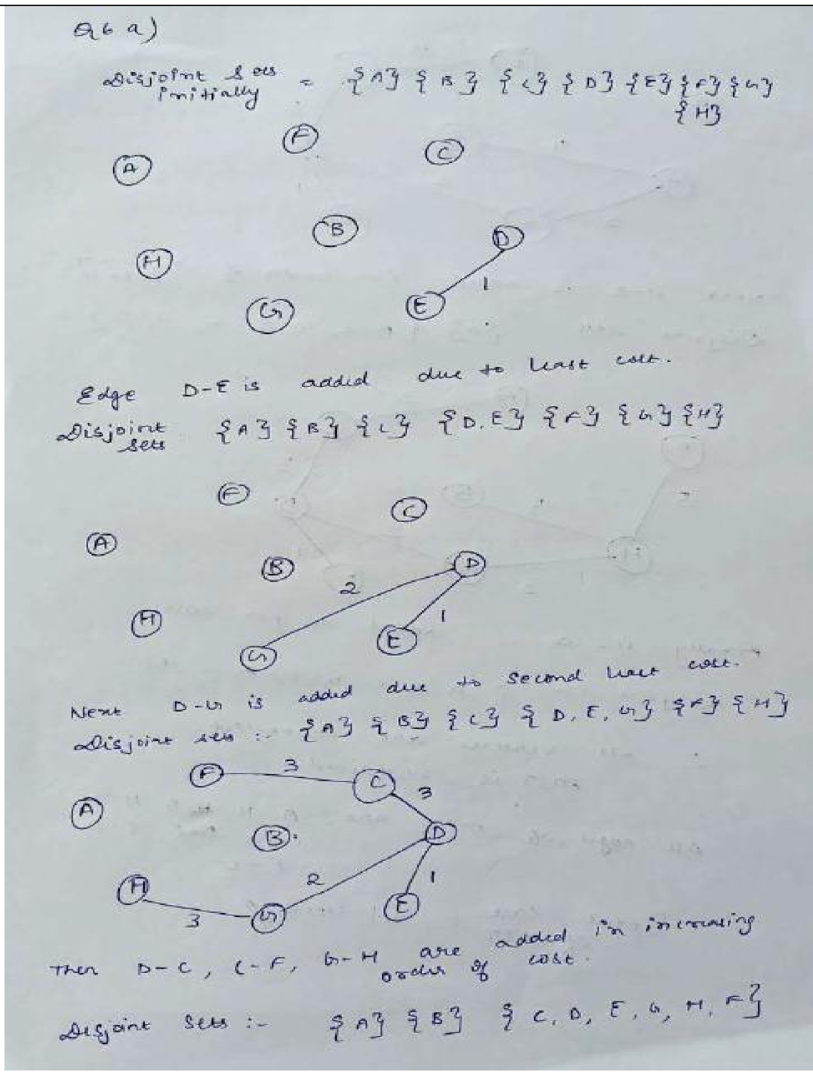
Final BFS Tree:

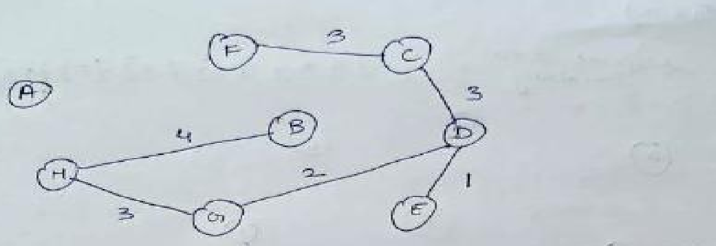


Scheme:

Computing BFS Tree from the given graph: 4 marks

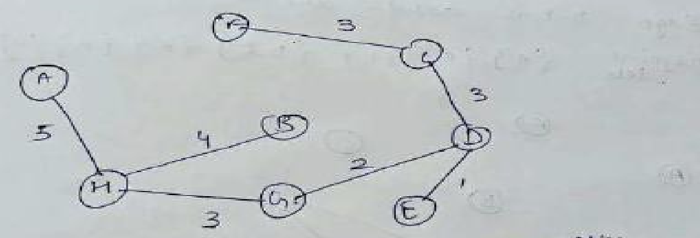
Partially correct answer: 2 marks

		Wrong answer: 0 marks	
6.	(a)	A weighted graph G = (V, E) below has the sets V = {A, B, C, D, E, F, G, H} of vertices and connected with certain number of edges as per the below diagram. Apply Kruskal algorithm on this graph to build its minimum spanning tree. Perform all steps of the main Kruskal's for-loop for the edges in increasing cost: (i) indicate the disjoint Sets (ii) If the edge is added, list all sets of vertices after their merging (iii) Finally, list all the edges forming the MST and give their total cost.  	[4]
	Solution	 Q6 a) Disjoint sets initially = {A} {B} {C} {D} {E} {F} {G} {H} Edge D-E is added due to least cost. Disjoint sets {A} {B} {C} {D, E} {F} {G} {H} Next D-G is added due to second least cost. Disjoint sets :- {A} {B} {C} {D, E, G} {F} {H} Then D-C, C-F, G-H are added in increasing order of cost. Disjoint sets :- {A} {B} {C, D, E, G, H, F}	



Next H-B is added in order of increasing cost.

Disjoint sets {A} {F, C, D, E, G, H, B}



Finally H-A is added as per cost.

Disjoint set {A, B, C, D, E, F, G, H}

∴ all vertices are connected

MST is obtained.

All edges of MST are: A-H, H-B, H-G, G-D, D-E, C-D, F-C

Total cost of MST = 21 units //

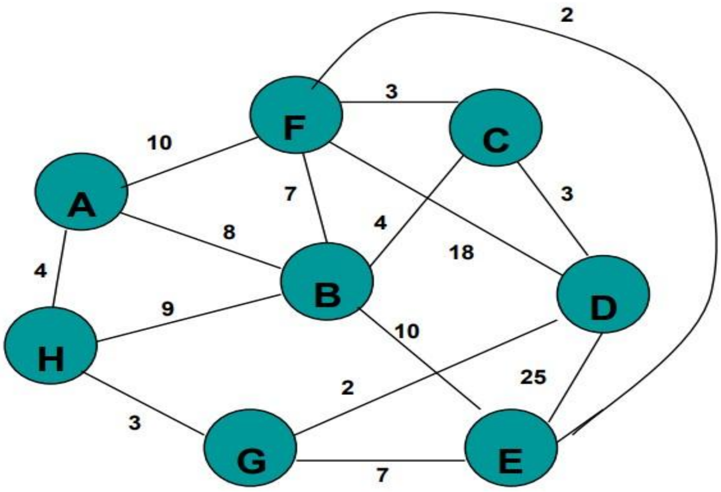
Scheme:

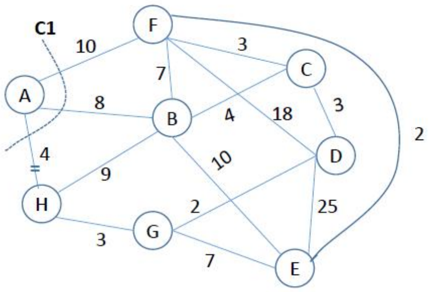
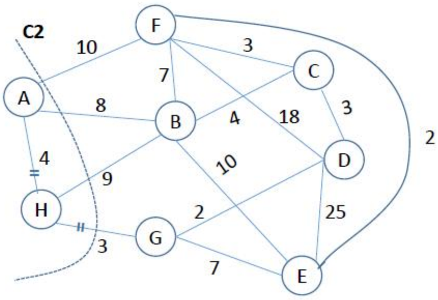
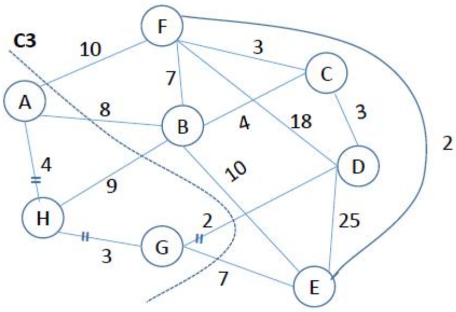
To find and draw step by step minimum spanning tree using Kruskal algorithm, indicating the disjoint sets, listing all sets of vertices after their merging, finding total mst cost --4 marks

Partially correct answer: 2 marks

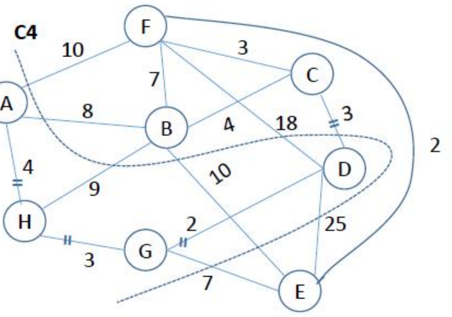
Wrong answer: 0 marks

(b) A weighted graph G = (V, E) below has the sets V = {A, B, C, D, E, F, G, H} of vertices and connected with certain number of edges as per the below diagram. Apply Prim's algorithm on this graph to build its minimum spanning tree. Compute the list all the edges forming the MST and give their total cost. [4]



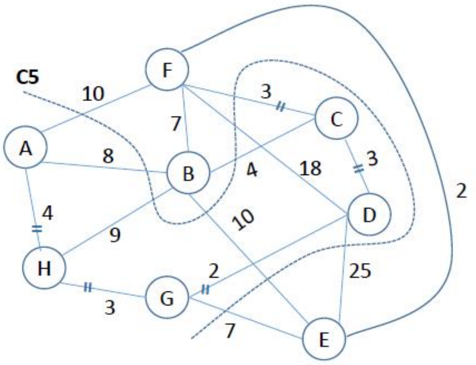
	Solution	**Let's assume the start vertex 'A'** **STEP-1**  **Visited Node: {A}** **Not Visited Node: {B,C,D,E,F,G,H}** **STEP-2**  **Visited Node: {A,H}** **Not Visited Node: {B,C,D,E,F,G}** **STEP-3**  **Visited Node: {A,H,G}** **Not Visited Node: {B,C,D,E,F}**	

STEP-4



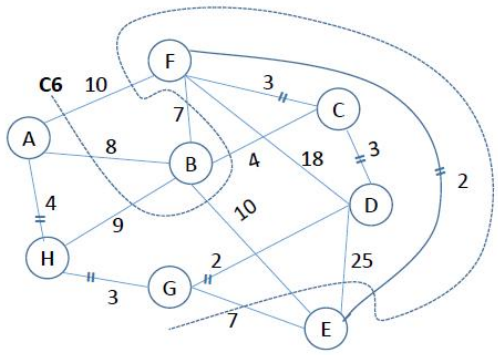
Visited Node: {A,H,G,D}
Not Visited Node: {B,C,E,F}

STEP-5



Visited Node: {A,H,G,D,C}
Not Visited Node: {B,E,F}

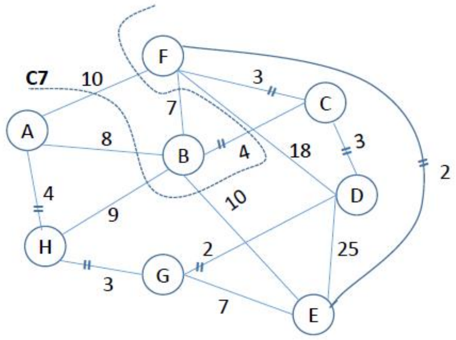
STEP-6



Visited Node: {A,H,G,D,C,F}
Not Visited Node: {B,E}

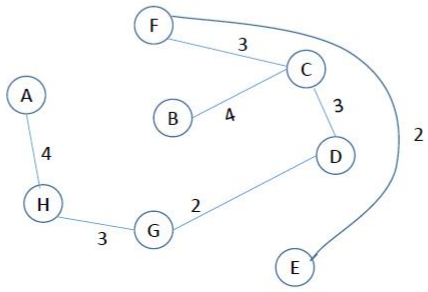
<table>
<tr><td></td><td></td><td>

STEP-7



Visited Node: {A,H,G,D,C,F,E}
Not Visited Node: {B}

MST:-



∴ Cost = 4+3+2+3+3+2+4 = 21

List of edges ={AH,HG,GD,DC,CF,BC,FE}

Scheme:

Using prim's algorithm on the given graph to build its minimum spanning tree --2 marks

For computing the list of all the edges forming the MST and finding total cost--2 marks

Partially correct answer: 2 marks

Wrong answer:0 marks

</td><td></td></tr>
<tr><td colspan="4">

SECTION-D

</td></tr>
<tr><td>7.</td><td>(a)</td><td>Assuming $n = 3^m$ with the integer $m = \log_3 n$, compute the time complexity of $T(n)$ by solving the recurrence $T(n) = 3T(n/3) + 6$ with the base condition $T(1) = 0$.</td><td>[4]</td></tr>
<tr><td></td><td>Solution</td><td>

Let $T(n)=T(3^m)=S(m)$, then $T\left(\frac{n}{3}\right)=T(3^{m-1})=S(m-1)$

Therefore, recurrence relation $T(n)=3T\left(\frac{n}{3}\right)+6$ can be expressed as, $S(m)=3S(m-1)+6$

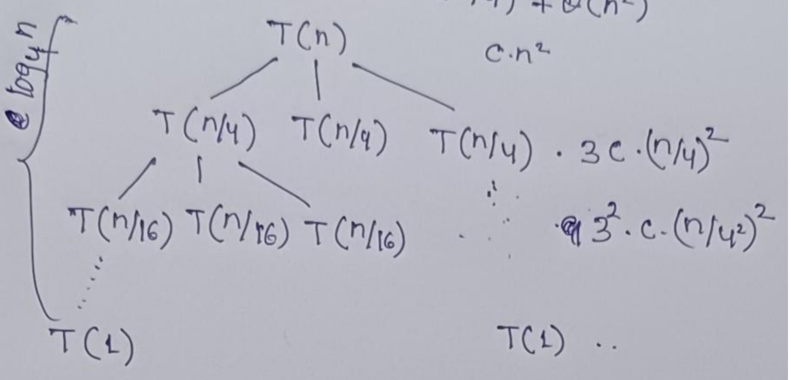
But, $S(m)=3S(m-1)+6=3^2S(m-2)+6\times3+6=\cdots=3^mS(0)+6\times(1+3+3^2\ldots3^{m-1})$

Also, $S(0)=T(1)=0$ and $6\times(1+3+3^2\ldots3^{m-1})=3(3^m-1)=3(n-1)$

Therefore, $S(m)=T(n)=3(n-1)$

Scheme:
Finding time complexity for the given recurrence equation: 4 marks
Partially correct answer: 2 marks
Wrong answer: 0 marks

</td><td></td></tr>
<tr><td></td><td>(b)</td><td></td><td>[4]</td></tr>
</table>

		Solve the recurrence using recurrence tree method $T(n) = 3T(n/4) + \Theta(n^2)$	
	Solution	$T(n) = 3T(n/4) + \Theta(n^2)$  Total Work done $= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 + \cdots$ $= cn^2\left[1 + \frac{3}{16} + \left(\frac{3}{16}\right)^2 + \left(\frac{3}{16}\right)^3\right.$ [∵ Using infinite GP Summation : $\frac{1}{1-r}$ ∵ r $= cn^2\left[\frac{1}{1-3/16}\right]$ **Scheme:** **Finding time complexity for the given recurrence equation: 4 marks** **Partially correct answer: 2 marks** **Wrong answer: 0 marks**	
8.	(a)	Solve the following recurrence relation using Master's theorem- $T(n) = 2T(n/4) + n^{0.51}$	[4]
	Solution	8. (a) Solve using Master's theorem: $T(n) = 2T(n/4) + n^{0.51}$ $a = 2,\ b = 4,\quad f(n) = n^{0.51}$ $= n^k \cdot \log^p n$ $k = 0.51,\ p = 0$ $\log_b a = \log_4 2$ $= 0.500 < k$ case 3: as $p = 0$, $\theta(n^{0.51} \cdot \log^0 n)$ $= \theta(n^{0.51})$	

<table>
<tr><td></td><td></td><td>

Scheme:

For finding time complexity using masters theorem: 4 Mark

Partially correct : 2 Mark can be awarded in case of minor mistakes.

Wrong Answer:0

</td><td></td></tr>
<tr><td></td><td>(b)</td><td>

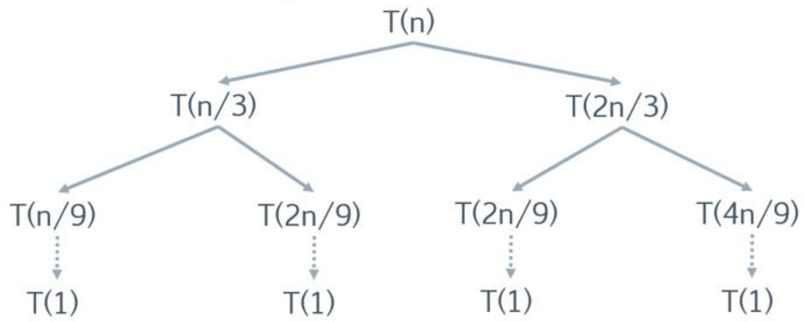
Solve the recurrence using recurrence tree method

T(n) = T(n/3) + T(2n/3) + O(n)

</td><td>[4]</td></tr>
<tr><td></td><td>Solution</td><td>

Recursion tree for T(n)=T(n/3)+T(2n/3)+cn

Recursion tree for T(n)

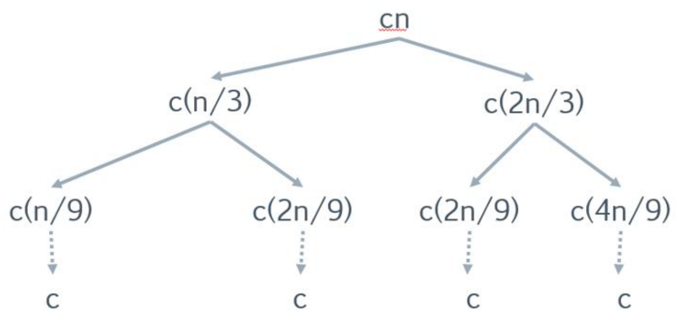


Shortest path will be most left one, because it operates on lowest value, and the most right one will be the longest one, that means tree is not balanced.

Shortest path can be define as: n–>(1/3)n–>$(1/3)^2$n–>...–>1

cn value on recursive tree:

Recursion tree for cn



Sum of each complete level is equal to cn.

Elements from shortest path are being divided by 3, so length of this path will be equal to **$\log_3 n$**. So if number of complete levels of recursion tree for shortest path is equal to **$\log_3 n$**, that means cost of algorithm for this path will be:
T(n)=cnlog3n=**Ω(nlogn)**

Scheme:

For drawing correct recursive tree and finding time complexity 4 Mark

Partially correct answer : 2 Mark can be awarded in case of minor mistakes.

Wrong Answer:0

</td><td></td></tr>
</table>