ABES Engineering College, Ghaziabad
B. Tech Odd Semester Make-Up Test

Printed Pages: 2
Session: 2022-2023

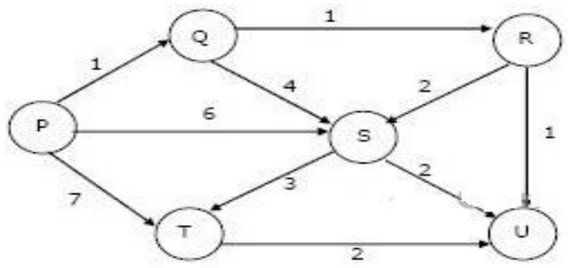
Course Code: KCS503 | **Roll No.:**
Course Name: DAA | **Date of Exam:**
Maximum Marks: | **Time:**

Instructions:

1. **Attempt All sections.**
2. **If require any missing data, then choose suitably.**

<table>
<tr><th>Q. No.</th><th>Question</th><th>Marks</th><th>CO</th><th>KL</th><th>PI</th></tr>
<tr><td colspan="6">Attempt All Questions   Total Marks: 10*10= 100</td></tr>
<tr><td>1a)</td><td>Write an algorithm for counting sort. Illustrate the operation of counting sort on the following array:
A={2,1,1,0,2,5,4,0,2,8,7,7,9,2,0,1,9}</td><td>5+5</td><td>CO1</td><td>K3</td><td>3.4.3</td></tr>
<tr><td>1b)</td><td>Solve the following recurrence relation:
(i) T(n)=T(n-1)+n4
(ii) T(n)=T(n/4) +T(n/2)+ n2</td><td>5+5</td><td>CO1</td><td>K3</td><td>3.4.3</td></tr>
<tr><td>2a)</td><td>Show that “If n ≥ 1, then for any n-key B-tree T of height h and minimum degree t ≥ 2, h ≤ log t (n+1)/2. Construct a B-tree of degree t=3 on the following data set and assume that B-Tree is initially empty.
<E,A, S, Y, Q, U, E, S, T, I, O, N, I, N, C></td><td>5+5</td><td>CO2</td><td>K3</td><td>4.2.1, 2.2.4</td></tr>
<tr><td>2b)</td><td>Prove that Red Black tree with ‘n’ internal nodes has height at most 2log2 (n+1). Show red back tree that result after successively inserting the keys <40, 50, 70, 30, 42, 15, and 20> into an initially empty red black tree.</td><td>5+5</td><td>CO2</td><td>K3</td><td>3.4.3, 2.4.1</td></tr>
<tr><td>3a)</td><td>(a) Find the maximum number of activities that are taken place for the given set of 10 activities:
A = (A1, A2, A3, A4, A5, A6, A7, A8, A9, A10)
Si = (1,2,3,4,7,8,9,9,11,12)
fi = (3,5,4,7,10,9,11,13,12,14)

(b) Find the optimal schedule for the following task with given weights (penalties) and deadlines.
<table>
<tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>d_i</td><td>4</td><td>2</td><td>4</td><td>3</td><td>1</td><td>4</td><td>6</td></tr>
<tr><td>w_i</td><td>70</td><td>60</td><td>50</td><td>40</td><td>30</td><td>20</td><td>10</td></tr>
</table></td><td>5+5</td><td>CO3</td><td>K3</td><td>2.3.2</td></tr>
</table>

<table>
<tr>
<td>3 b)</td>
<td>Suppose we run Dijkstra’s single source shortest-path algorithm on the following edge-weighted directed graph with vertex P as the source.


In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized? [GATE2004]</td>
<td>10</td>
<td>CO3</td>
<td>K3</td>
<td>2.3.2</td>
</tr>
<tr>
<td>4a)</td>
<td>Illustrate the Floyd’s-Warshall algorithm for all pairs shortest path problem. Apply the same to the following weight matrix.

$$\begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

</td>
<td>5+5</td>
<td>CO4</td>
<td>K3</td>
<td>4.3.1</td>
</tr>
<tr>
<td>4b)</td>
<td>Consider the following four matrices.
<table>
<tr><th>Matrix</th><th>Order</th></tr>
<tr><td>A1</td><td>4 X 10</td></tr>
<tr><td>A2</td><td>10 X 3</td></tr>
<tr><td>A3</td><td>3 X 12</td></tr>
<tr><td>A4</td><td>12 X 20</td></tr>
<tr><td>A5</td><td>20 X 7</td></tr>
</table>
i. How many different possible sequences can be generated to multiply above given matrices?
ii. Find the optimal parenthesization of matrix chain multiplication.</td>
<td>2+8</td>
<td>CO4</td>
<td>K3</td>
<td>4.3.1</td>
</tr>
<tr>
<td>5a)</td>
<td>Write short notes on the following:
i. NP Completeness
i. Randomization
ii. Approximation</td>
<td>5+5</td>
<td>CO5</td>
<td>K2</td>
<td>3.2.1</td>
</tr>
<tr>
<td>5b)</td>
<td>Given a string 'T' and pattern 'P' as follows:
T = bacbabababaca, P = ababaca
Compute the KMP Algorithm to find whether pattern 'P' occurs in string 'T'.</td>
<td>10</td>
<td>CO5</td>
<td>K3</td>
<td>3.2.1</td>
</tr>
</table>

CO Course Outcomes mapped with respective question

KL Bloom's knowledge Level (K1, K2, K3, K4, K5, K6)

K1- Remember, K2- Understand, K3-Apply, K4- Analyze, K5: Evaluate, K6- Create

ABES Engineering College, Ghaziabad

B. Tech Odd Semester Make-Up Test Solution

Ques 1(a) Write an algorithm for counting sort. Illustrate the operation of counting sort on the following array: A={2,1,1,0,2,5,4,0,2,8,7,7,9,2,0,1,9}

Ans:

```
COUNTING-SORT(A, B, k)
let C[0..k] be a new array
for i = 0 to k
    C[i] = 0
for j = 1 to A.length
    C[A[j]] = C[A[j]] + 1
// C[i] now contains the number of elements equal to i.
for i = 1 to k
    C[i] = C[i] + C[i - 1]
// C[i] now contains the number of elements less than or equal to i.
for j = A.length downto 1
    B[C[A[j]]] = A[j]
    C[A[j]] = C[A[j]] - 1
```

[5 marks]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A =	2	1	1	0	2	5	4	0	2	8	7	7	9	2	0	1	9

Here Tot. no. of elements = 17, (n)   max value K = 9.

→ Create an array of length (K+1), count array and initialize as zero, And an output array with same length of Array A.

	0	1	2	3	4	5	6	7	8	9
Count	0	0	0	0	0	0	0	0	0	0

→ Start pointer from 0 in Array A and count the frequency of each number and update the count array.

for(i=0; i<n; i++)

{ Count [i] = Count [A[i]]+1 }

	0	1	2	3	4	5	6	7	8	9
Count	3	3	4	0	1	1	0	2	1	2

→ Now update count array by adding its previous value.

for (i=0; i<=K; i++)

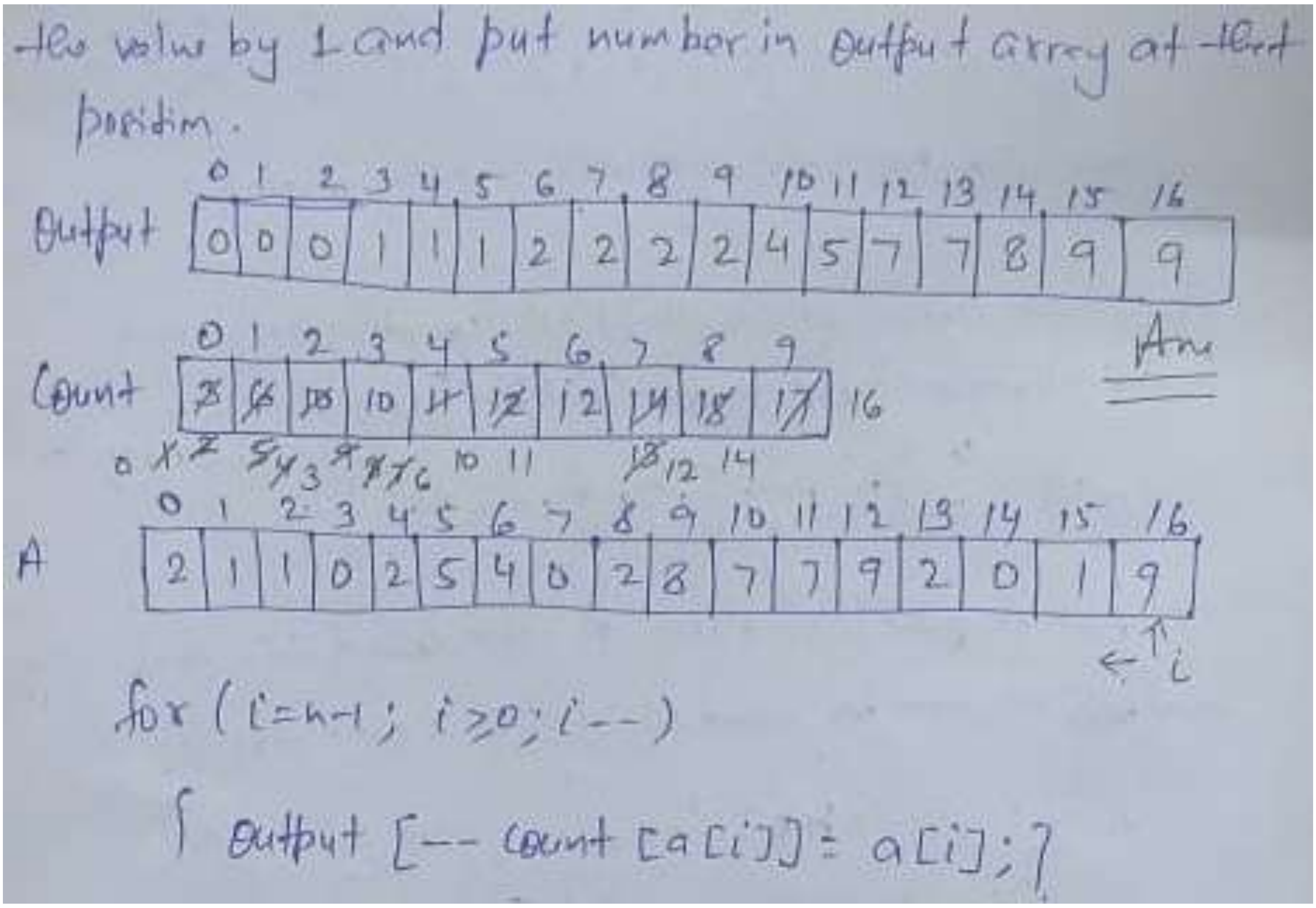
{ Count [i] = Count [i]+ Count [i-1] ; }

	0	1	2	3	4	5	6	7	8	9
Count	3	6	10	10	11	12	12	14	15	17

→ this should be same as n.

→ Now start pointer from last index of array A and start reading value.

→ Check the value of count array in that index and subtract



[5 marks]

Ques 1(b) Solve the following recurrence relation:

(ii) T(n)=T(n/4) +T(n/2)+ n2 (i) T(n)=T(n-1)+n4

Solve the recurrence relation

$$T(n) = T(n-1) + n^4 \longrightarrow ①$$

by iteration method

$$T(n) = [T(n-2) + (n-1)^4] + n^4 \quad \left\{ \begin{array}{l} \text{by replacing} \\ T(n-1) \end{array} \right\}$$

[1 Mark]

$$= T(n-2) + (n-1)^4 + n^4$$

$$= [T(n-3) + (n-2)^4] + (n-1)^4 + n^4 \quad \left\{ \begin{array}{l} \text{by replacing} \\ T(n-2) \end{array} \right\}$$

$$T(n) = T(n-3) + (n-2)^4 + (n-1)^4 + n^4$$

[2 Marks]

$$\vdots$$

$$T(n) = T(n-k) + (n-k+1)^4 + (n-k+2)^4 + \cdots + (n-1)^4 + n^4 \longrightarrow ②$$

if $k = n$, then

[3 Marks]

$$T(n) = T(n-n) + (n-n+1)^4 + (n-n+2)^4 + \cdots + (n-1)^4 + n^4$$

$$= T(0) + 1^4 + 2^4 + 3^4 + \cdots + (n-1)^4 + n^4$$

$$= T(0) + \sum_{i=1}^{n} n^4$$

$$= T(0) + n^4 \cdot \sum_{i=1}^{n} 1$$

$$= T(0) + n^4 \cdot n \quad \left\{ \sum_{i=1}^{n} 1 = n \right\}$$

$$= 0 + n^5$$

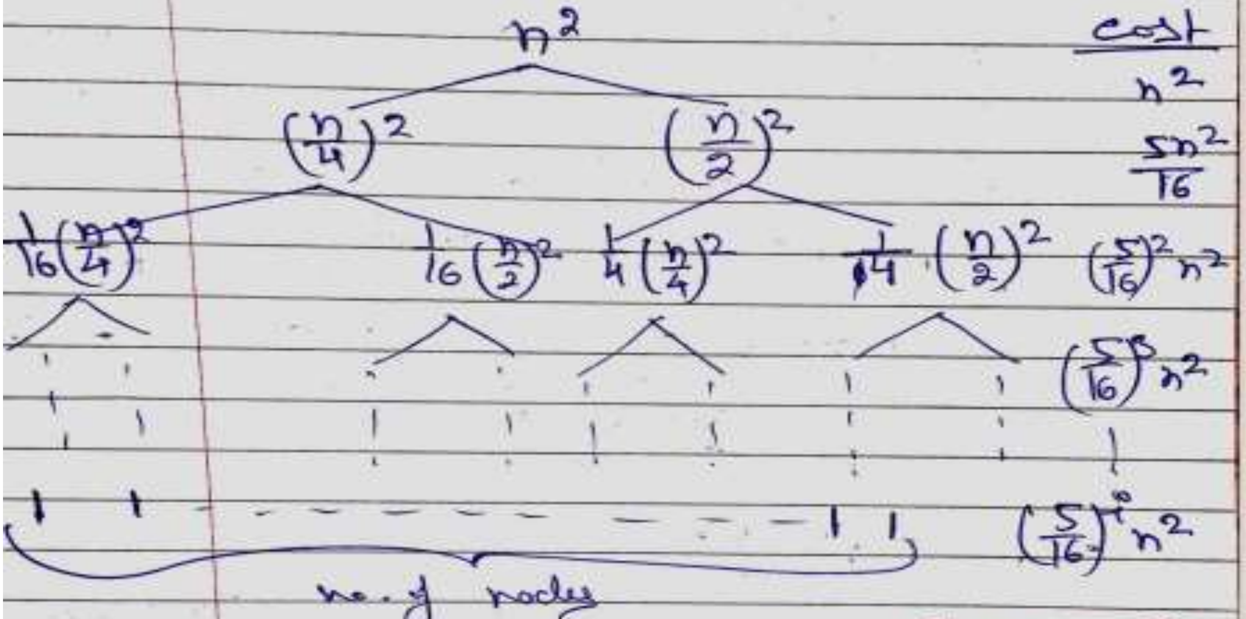
i.e. $\boxed{T(n) = \theta(n^5)}$ $\left\{ \begin{array}{l} \text{assume } T(0) = 0 \\ \text{as initial condition} \end{array} \right\}$

[5 Marks]

(b)(ii) :- Solve the recurrence relation

Solution :- $T(n) = T(n/4) + T(n/2) + n^2 \longrightarrow (1)$

Recursion Tree for the given relation is



[2 Marks]

⇒ First, we have the find the height of this recursion tree.

By observation we got that the node on the right side have the longest path of the tree. Hence the node at depth 'i' reflects a subproblem of size $\frac{n^2}{2^i}$

i.e. the subproblem size hits n=1, when $\frac{n^2}{2^i} = 1$ i.e. $n^2 = 2^i$

taking log on both side

$$\log n^2 = \log 2^i$$

$$2\log n = i$$

or we can say that the height of the tree will be $\log_2 n$. [3 Marks]

i.e. the total cost of the tree is

$$T(n) = n^2 + \frac{5}{16}n^2 + \left(\frac{5}{16}\right)^2 n^2 + \cdots \text{ till height}$$

$$= n^2\left[1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \left(\frac{5}{16}\right)^3 + \cdots + \left(\frac{5}{16}\right)^{\log n}\right]$$

$$= n^2\left[\sum_{i=0}^{\log_2 n}\left(\frac{5}{16}\right)^i\right]$$

$$T(n) = n^2\left[\frac{\left(\frac{5}{16}\right)^{\log n} - 1}{\left(\frac{5}{16} - 1\right)}\right] \longrightarrow (2)$$

[4 Marks]

The above eq^n is very complicated, so use infinite geometric series as an upper bound. Hence the new eq^n will be

$$T(n) = n^2 \sum_{i=0}^{\infty}\left(\frac{5}{16}\right)^i$$

$$= n^2\left[\frac{1}{1 - \frac{5}{16}}\right]$$

$$= n^2\left[\frac{16}{11}\right]$$

$\boxed{T(n) = O(n^2)}$ after neglecting constant.

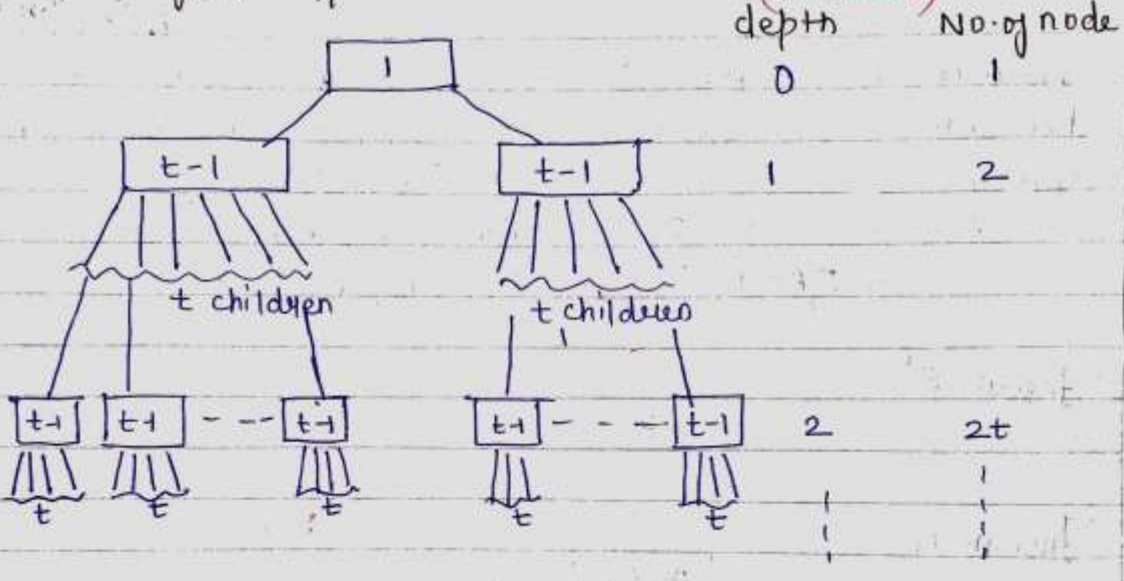
[5 Marks]

Ques 2(a) Show that "If n ≥ 1, then for any n-key B-tree T of height h and minimum degree t ≥ 2, h ≤ log t (n+1)/2. Construct a B-tree of degree t=3 on the following data set and assume that B-Tree is initially empty.

<E,A, S, Y, Q, U, E, S, T, I, O, N, I, N, C>

a) part Show that if $n \geq 1$, then for any n- Key B Tree T of height h and minimum degree $t \geq 2$, $h \leq \log_t (n+1)/2$ (5 marks)

Sol



no. of key $n \geq 1 + (t-1) \sum_{i=1}^{h} 2t^{i-1}$

$$n \geq 1 + 2(t-1)\left[1 + t + t^2 + --- + t^{h-1}\right)$$

$$\geq 1 + 2(t-1)\left(\frac{t^h - 1}{(t-1)}\right)$$

$$n \geq 1 + 2(t^h - 1)$$

$$n \geq 2t^h - 1$$

$$n \geq 2t^h - 1$$

$$n + 1 \geq 2t^h$$

$$\left(\frac{n+1}{2}\right) \geq t^h$$

taking log on both side

$$h \log t \leq \log\left(\frac{n+1}{2}\right)$$

$$\boxed{h \leq \log_t\left(\frac{n+1}{2}\right)}$$

[5 Marks]

Construct a B-Tree of degree t=3 on the following dataset and assume that B Tree is initially empty.

5 marks

$t = 3$ so $min = 2$, $max = 5$

< E, A, S, Y, Q, U, E, S, T, I, O, N, I, N, C >

Insert 'E'

$\boxed{E}$

Insert A

$\boxed{A \mid E}$

Insert S

$\boxed{A \mid E \mid S}$

1 marks

Insert Y

$\boxed{A \mid E \mid S \mid Y}$

Insert Q.

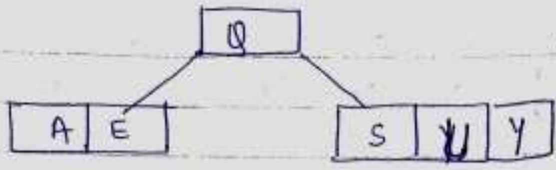
$\boxed{A \mid E \mid Q \mid S \mid Y}$

1 marks

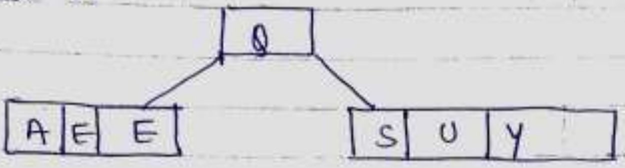
2 marks

Insert U

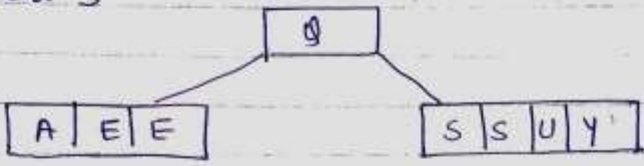
As Max capacity will exceed if we insert U' so first split then insert.



~~In~~
Insert 'E'

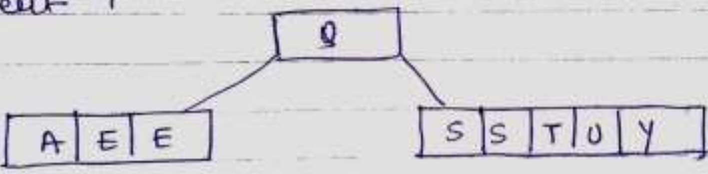


Insert S

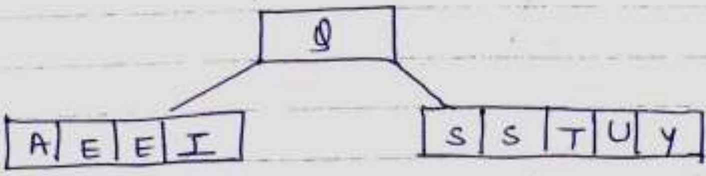


3marks

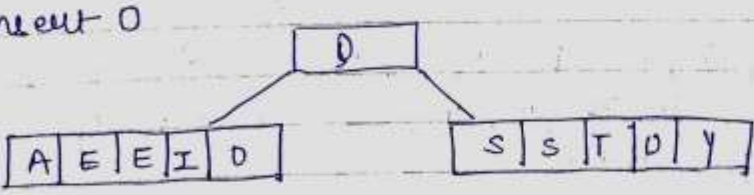
Insert T



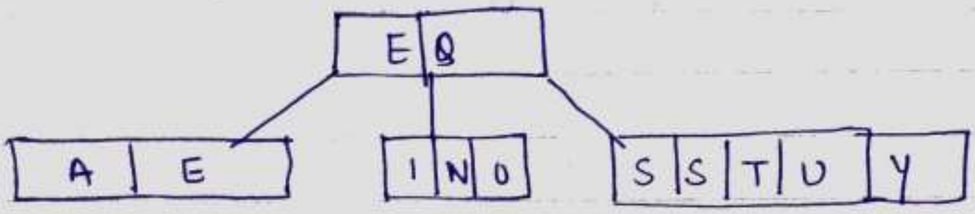
Insert I



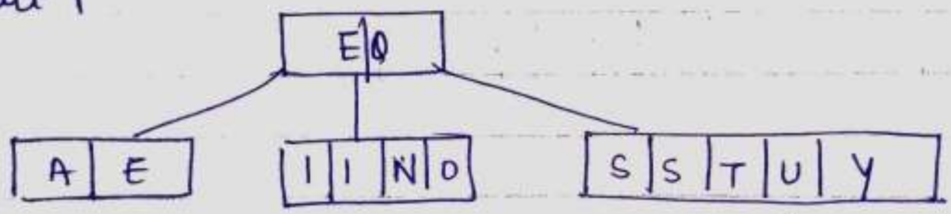
Insert O



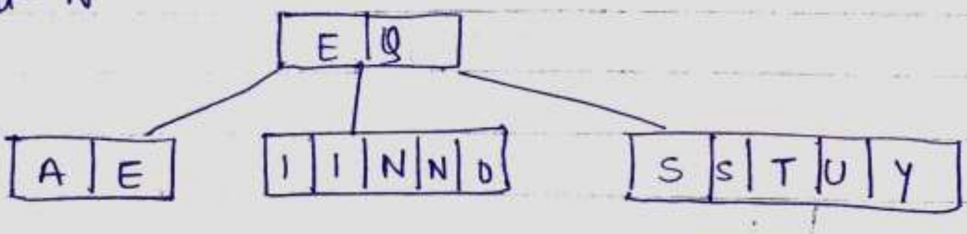
Insert 'N' When we insert 'N' it will voilate the property. So first split then insert.



Insert I

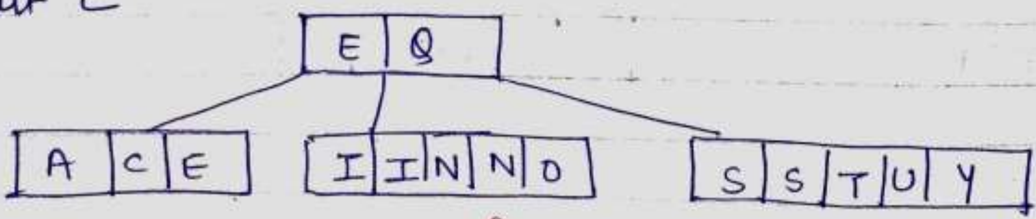


Insert 'N'



5 marks

Insert C



final tree

Ques 2(b) Prove that Red Black tree with n internal nodes has height at most 2log2 (n+1). Show red back tree that result after successively inserting the keys **<40, 50, 70, 30, 42, 15, and 20>** into an initially empty red black tree.

a) part Prove that red black tree with 'n' internal nodes has height atmost $2\log_2(n+1)$. — 5 marks

The Sub tree rooted at any node 'x' contains atleast $2^{bh(x)}-1$ internal nodes

The node 'x' with any two children, each child is having $bh(x)$ or $bh(x)-1$ height

Then for each child has atleast

$$2^{bh(x)-1}-1 \text{ internal node.}$$

To calculate the total no. of internal node of whole tree (It can be written as Sum of Left child + Right child + root node

$$(2^{bh(x)-1}-1) + (2^{bh(x)-1}-1) + 1$$

$$= 2^{bh(x)-1}-1 \quad \text{— (1)}$$

Let 'h' is the height of RB tree
So $bh(x) = h/2$
Put this in eq (1)

$$n \geq 2^{h/2}-1$$

$$n+1 \geq 2^{h/2}$$

Taking log both Sides

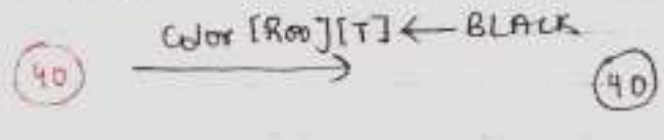
$\log(n+1) \geq h/2$

$\boxed{2 \log(n+1) \geq h}$ — Hence proved

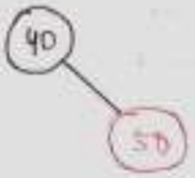
[5 Marks]

Show that red black tree that after successively inserting the keys 40, 50, 70, 30, 42, 15, 20 into an initially empty red black tree. — 5 marks

Insert (40)

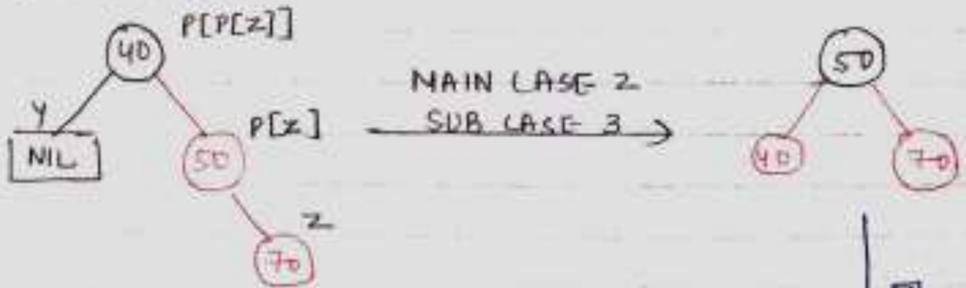


Insert (50)

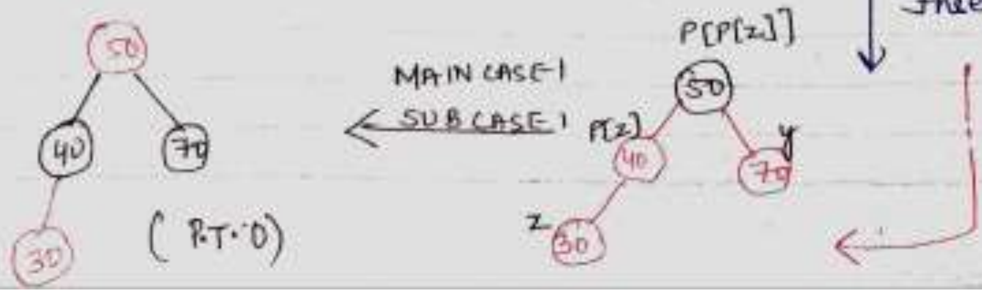


1 Mark

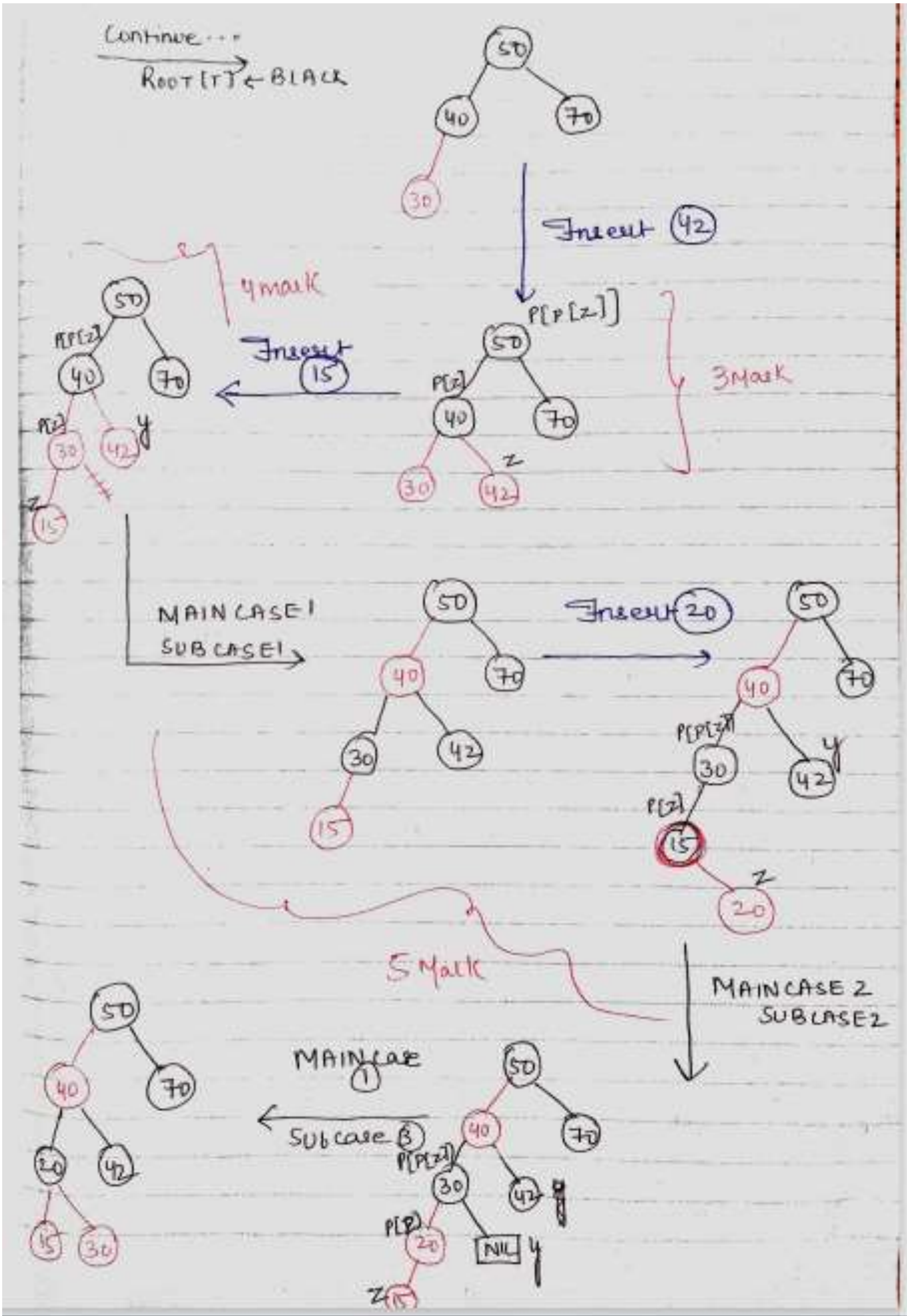
Insert (70)



Insert (30)



2 Mark



Continue ...
ROOT[T] ← BLACK
Insert 42
P[P[z]]
P[z]
z
3Mark
Insert 15
4mark
P[P[z]]
P[z]
y
z
MAINCASE1
SUBCASE1
Insert 20
P[P[z]]
P[z]
y
z
5Mark
MAINCASE 2
SUBCASE2
MAINcase 1
Subcase B
P[P[z]]
P[z]
NIL
y
z

Ques 3(a)

(i) Find the maximum number of activities that are taken place for the given set of 10 activities:

A = (A1, A2, A3, A4, A5, A6, A7, A8, A9, A10)

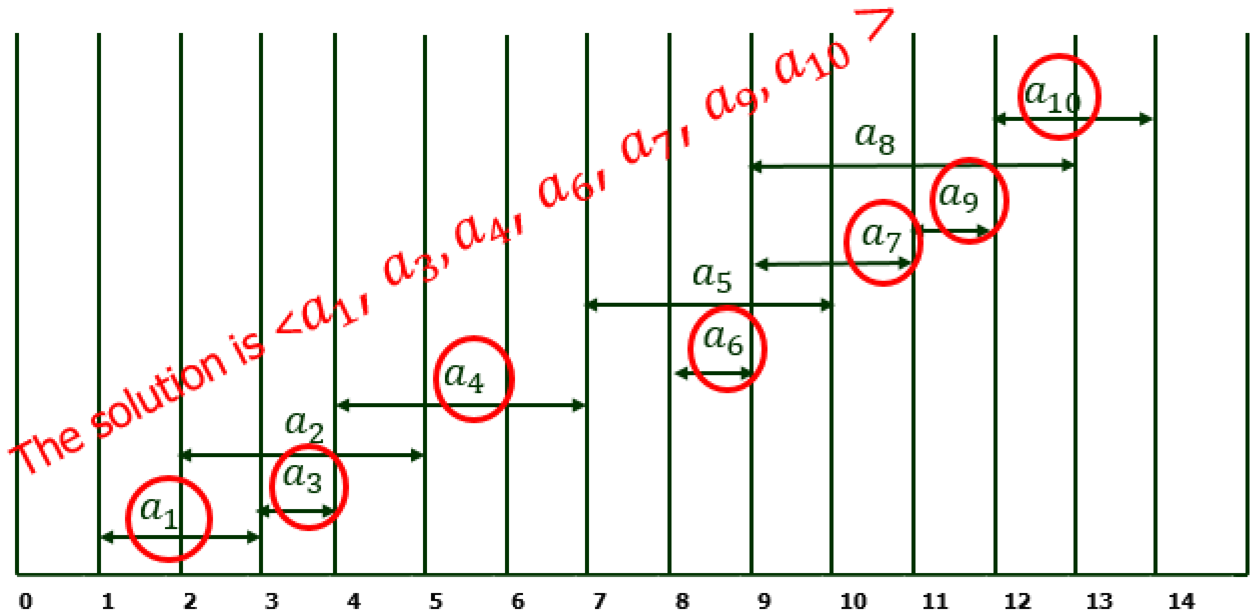
Si = (1,2,3,4,7,8,9,9,11,12)

fi = (3,5,4,7,10,9,11,13,12,14)

Activity	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
s_i	1	2	3	4	7	8	9	9	11	12
f_i	3	5	4	7	10	9	11	13	12	14

First arranging the following activities in increasing order on their finishing

Activity	a_1	a_3	a_2	a_4	a_6	a_5	a_7	a_9	a_8	a_{10}
s_i	1	3	2	4	8	7	9	11	9	12
f_i	3	4	5	7	9	10	11	12	13	14



[5 Marks]

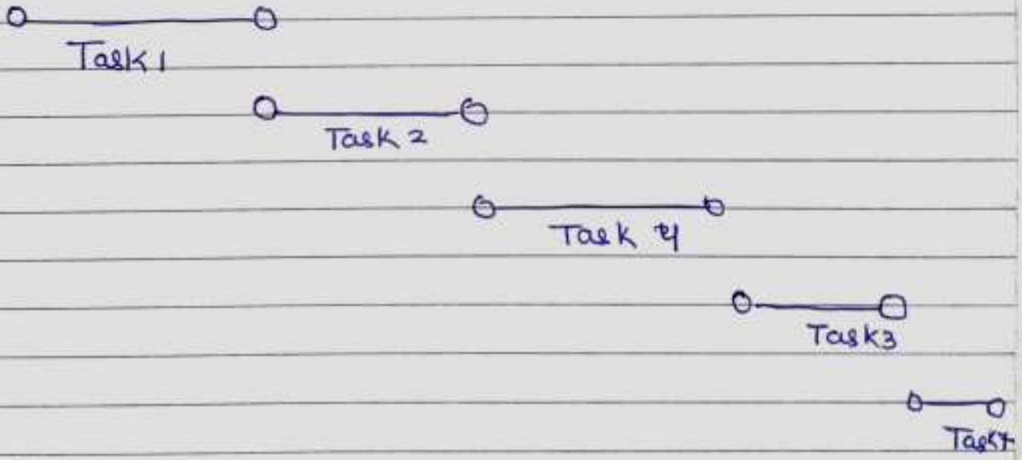
(ii)

Q.3 a) b part

i	1	2	3	4	5	6	7
di	4	2	4	3	1	4	6
wi	70	60	50	40	30	20	10

Sol^n Penalties already in descending order

First we select task 1, then select 2, then select task 4, then select task 3, then select task 7

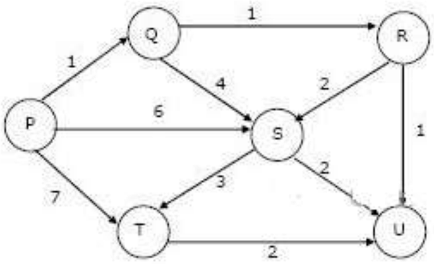


Sol^n → < 1, 2, 4, 3, 7 >

Penalty = 30 + 20 = 50 Ans

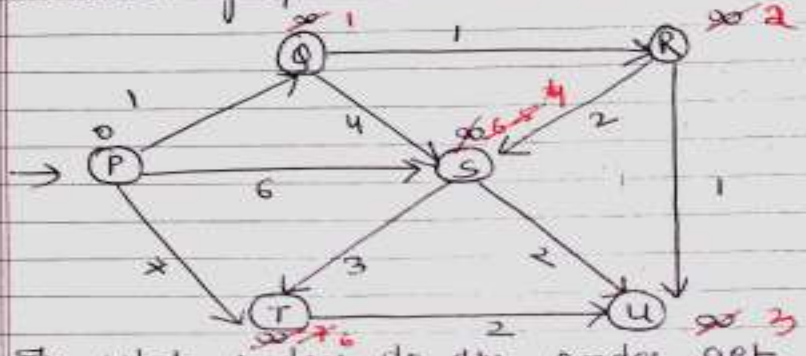
[5 Marks]

Ques 3(b) Suppose we run Dijkstra's single source shortest-path algorithm on the following edge-weighted directed graph with vertex P as the source.



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized? **[GATE2004]**

Que 3(b) Suppose we run Dijkstra Single Source Shortest path algorithm on the following edge-weighted directed graph with vertex P as the Source



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized ?

	P	Q	R	S	T	U
Key	0	∞	∞	∞	∞	∞
π[v]	NIL	NIL	NIL	NIL	NIL	NIL

	$\boxed{P}$	Q	R	S	T	U
Key	0	1	∞	6	7	∞
π[v]	NIL	P	NIL	P	P	NIL

	$\boxed{P}$	$\boxed{Q}$	R	S	T	U
Key	0	1	2	5	7	∞
π[v]	NIL	P	Q	Q	P	NIL

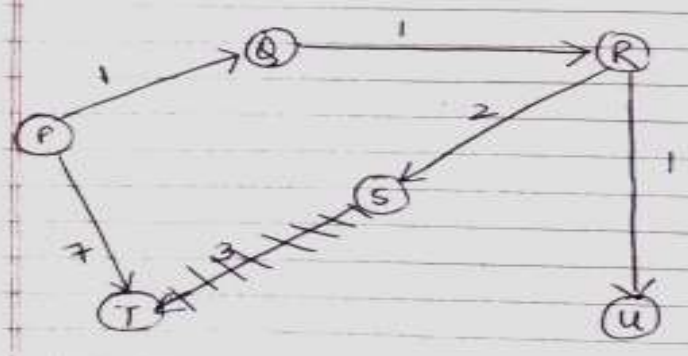
	$\boxed{P}$	$\boxed{Q}$	$\boxed{R}$	S	T	U
Key	0	1	1	4	7	3
π[v]	NIL	P	Q	R	P	R

P	Q	R	S	T	U
0	1	1	4	7	3
NIL	P	Q	R	P	R

P	Q	R	S	T	U
0	1	1	4	7	3
NIL	P	Q	R	P	R

P	Q	R	S	T	U
0	1	1	3	7	3
NIL	P	Q	R	P	R

Final Path :-



Order :- P → Q → R → U → S → T

—— 10 Marks

Sol (4-a): Illustrate the Floyd's-Warshall algorithm for all pairs shortest path problem. Apply the same to the following weight matrix.

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & 3 & 8 & \infty & -4 \\
\infty & 0 & \infty & 1 & 7 \\
\infty & 4 & 0 & \infty & \infty \\
2 & \infty & -5 & 0 & \infty \\
\infty & \infty & \infty & 6 & 0
\end{bmatrix}
\end{array}
$$

Ans:

```
FLOYD-WARSHALL(W)
n = W.rows
D^(0) = W
for k = 1 to n
    let D^(k) = (d_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
            d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
return D^(n)
```

[5 marks]

Que. 4(a)

$$D^0 = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^1 = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$ when $K=1$

* $d_{23}^{(1)} \leftarrow \min(d_{23}^{(0)}, d_{21}^{(0)} + d_{13}^{(0)})$

$\leftarrow \min(\infty, \infty+8)$

$\leftarrow \min(\infty)$

* $d_{24}^{(1)} \leftarrow \min(d_{24}^{(0)}, d_{21}^{(0)} + d_{14}^{(0)})$

$\leftarrow \min(1, \infty+\infty)$

$\leftarrow \min(1)$

* $d_{25}^{(1)} \leftarrow \min(d_{25}^{(0)}, d_{21}^{(0)} + d_{15}^{(0)})$

$\leftarrow \min(7, \infty + -4) = \min(7)$

* $d_{32} \leftarrow \min(d_{32}^{(0)}, d_{31}^{(0)} + d_{12}^{(0)})$

$\leftarrow \min(4, \infty+3) \leftarrow \min(4)$

* $d_{42}^{(1)} \leftarrow \min(d_{42}^{(0)}, d_{41}^{(0)} + d_{12}^{(0)})$

$\leftarrow \min(\infty, 2+3)$

$\leftarrow \min(5)$

* $d_{45}^{(1)} \leftarrow \min(d_{45}^{(0)} + d_{41}^{(0)} + d_{15}^{(0)})$

$\leftarrow \min(\infty, 2+(-4))$

$d_{45}^{(1)} \leftarrow -2$

$$D^2 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

$$D^3 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

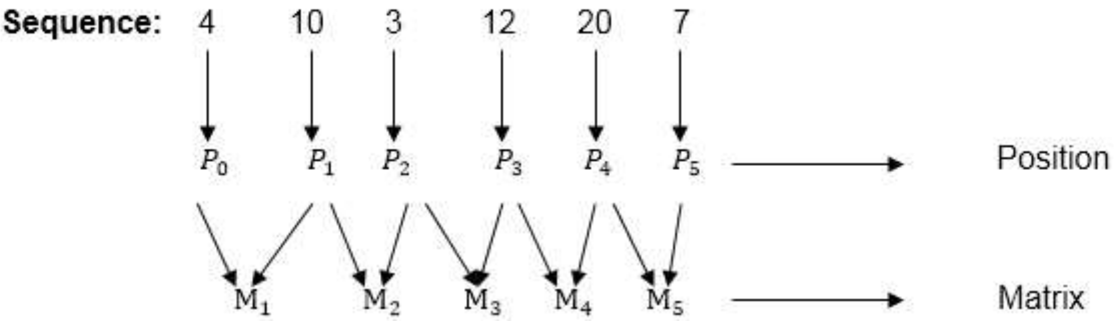
$$D^4 = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix} \end{array}$$

[5 Marks]

Sol (4-b): We are given the sequence {4, 10, 3, 12, 20, and 7}. The matrices have size 4 x 10, 10 x 3, 3 x 12, 12 x 20, 20 x 7. We need to compute M [i,j], 0 ≤ i, j≤ 5. We know M [i, i] = 0 for all i.

1	2	3	4	5	
0					1
	0				2
		0			3
			0		4
				0	5

Let us proceed with working away from the diagonal. We compute the optimal solution for the product of 2 matrices.



Here P_0 to P_5 are Position and M_1 to M_5 are matrix of size (p_i to p_{i-1})

On the basis of sequence, we make a formula

For M_i ⟶ p [i] as column

p [i-1] as row

Step-1: In Dynamic Programming, initialization of every method done by '0'. So we initialize it by '0'. It will sort out diagonally.

M[1,1]= 0, M[2,2]= 0, M[3,3]= 0, M[4,4]= 0, M[5,5]= 0

Step-2: Calculation of Product of 2 matrices:

1. M[1,2] = min{ M[1,1] + M[2,2]+ $p_0p_1p_2$
   = min{0 + 0 + 4 x 10 x 3
   = 120
2. M[2,3] = min{ M[2,2] + M[3,3]+ $p_1p_2p_3$
   = min{0 + 0 + 10 x 3 x 12
   = 360
3. M[3,4] = min{ M[3,3] + M[4,4]+ $p_2p_3p_4$
   = min{0 + 0 + 3 x 12 x 20
   = 720
4. M[4,5] = min{ M[4,4] + M[5,5]+ $p_3p_4p_5$
   = min{0 + 0 + 12 x 20 x 7
   = 1680

1	2	3	4	5	
0	120				1
	0	360			2
		0	720		3
			0	1680	4
				0	5

- After that second diagonal is sorted out and we get all the values corresponded to it

Step-3: Now the third diagonal will be solved out in the same way.
Now product of 3 matrices:
1. M [1, 3] = M_1 M_2 M_3
    I. There are two cases by which we can solve this multiplication: (M_1 x M_2) + M_3, M_1+ (M_2x M_3)
    II. After solving both cases we choose the case in which minimum output is there.

$$M\ [1,3] = \min \begin{Bmatrix} M\ [1,2] + M\ [3,3] + p_0\, p_2 p_3 = 120 + 0 + 4.3.12 \ = \ 264 \\ M\ [1,1] + M\ [2,3] + p_0\, p_1 p_3 = 0 + 360 + 4.10.12 \ = \ 840 \end{Bmatrix}$$

M [1, 3] =264
As Comparing both output **264** is minimum in both cases so we insert **264** in table and (M_1 x M_2) + M_3 this combination is chosen for the output making.

2. M [2, 4] = M_2 M_3 M_4
    I. There are two cases by which we can solve this multiplication: (M_2x M_3)+M_4, M_2+(M_3 x M_4)
    II. After solving both cases we choose the case in which minimum output is there.

$$M\ [2,4] = \min \begin{Bmatrix} M[2,3] + M[4,4] + \ p_1 p_3 p_4 = 360 + 0 + 10.12.20 = 2760 \\ M[2,2] + \ M[3,4] + \ p_1 p_2 p_4 = 0 + 720 + 10.3.20 = \ 1320 \end{Bmatrix}$$

M [2, 4] = 1320
As Comparing both output **1320** is minimum in both cases so we insert **1320** in table and M_2+(M_3 x M_4) this combination is chosen for the output making.

3. M [3, 5] = M_3 M_4 M_5
    I. There are two cases by which we can solve this multiplication: (M_3 x M_4) + M_5, M_3+ (M_4xM_5)
    II. After solving both cases we choose the case in which minimum output is there.

$$M\ [3,5] = \min \begin{Bmatrix} M[3,4] + M[5,5] + p_2 p_4 p_5 = 720 + 0 + 3\ .20.7 \ = \ 1140 \\ M[3,3] + \ M\ [4,5] + \ p_2 p_3 p_5 = 0 + 1680 + 3.12.7 = 1932 \end{Bmatrix}$$

M [3, 5] = 1140
As Comparing both output **1140** is minimum in both cases so we insert **1140** in table and (M_3 x M_4) + M_5this combination is chosen for the output making.

1	2	3	4	5	
0	120				1
	0	360			2
		0	720		3
			0	1680	4
				0	5

1	2	3	4	5	
0	120	264			1
	0	360	1320		2
		0	720	1140	3
			0	1680	4
				0	5

Step-4: Now Product of 4 matrices:
M [1, 4] = M_1 M_2 M_3 M_4
There are three cases by which we can solve this multiplication:

1. (M_1 x M_2 x M_3) M_4
2. M_1 x(M_2 x M_3 x M_4)
3. (M_1 xM_2) x (M_3 x M_4)

1. After solving these cases we choose the case in which minimum output is there

$$M\,[1,4] = \min \begin{cases} M[1,3] + M[4,4] + p_0p_3p_4 = 264 + 0 + 4.12.20 = 1224 \\ M[1,2] + M[3,4] + p_0p_2p_4 = 120 + 720 + 4.3.20 = 1080 \\ M[1,1] + M[2,4] + p_0p_1p_4 = 0 + 1320 + 4.10.20 = 2120 \end{cases}$$

M [1, 4] =1080

As comparing the output of different cases then '**1080**' is minimum output, so we insert 1080 in the table and (M_1 xM_2) x (M_3 x M_4) combination is taken out in output making,

2. M [2, 5] = M_2 M_3 M_4 M_5

There are three cases by which we can solve this multiplication:

1. (M_2 x M_3 x M_4)x M_5
2. M_2 x(M_3 x M_4 x M_5)
3. (M_2 x M_3)x (M_4 x M_5)

After solving these cases we choose the case in which minimum output is there

$$M\,[2,5] = \min \begin{cases} M[2,4] + M[5,5] + p_1p_4p_5 = 1320 + 0 + 10.20.7 = 2720 \\ M[2,3] + M[4,5] + p_1p_3p_5 = 360 + 1680 + 10.12.7 = 2880 \\ M[2,2] + M[3,5] + p_1p_2p_5 = 0 + 1140 + 10.3.7 = 1350 \end{cases}$$

M [2, 5] = 1350

As comparing the output of different cases then '**1350**' is minimum output, so we insert 1350 in the table and M_2 x(M_3 x M_4 xM_5)combination is taken out in output making.

1	2	3	4	5	
0	120	264			1
	0	360	1320		2
		0	720	1140	3
			0	1680	4
				0	5

→

1	2	3	4	5	
0	120	264	1080		1
	0	360	1320	1350	2
		0	720	1140	3
			0	1680	4
				0	5

Step-5: Now Product of 5 matrices:

M [1, 5] = M_1 M_2 M_3 M_4 M_5

There are five cases by which we can solve this multiplication:

1. (M_1 x M_2 xM_3 x M_4)x M_5
2. M_1 x(M_2 xM_3 x M_4 xM_5)
3. (M_1 x M_2 xM_3)x M_4 xM_5
4. M_1 x M_2x(M_3 x M_4 xM_5)

After solving these cases we choose the case in which minimum output is there

$$M\,[1,5] = \min \begin{cases} M[1,4] + M[5,5] + p_0p_4p_5 = 1080 + 0 + 4.20.7 = 1544 \\ M[1,3] + M[4,5] + p_0p_3p_5 = 264 + 1680 + 4.12.7 = 2016 \\ M[1,2] + M[3,5] + p_0p_2p_5 = 120 + 1140 + 4.3.7 = 1344 \\ M[1,1] + M[2,5] + p_0p_1p_5 = 0 + 1350 + 4.10.7 = 1630 \end{cases}$$

M [1, 5] = 1344

As comparing the output of different cases then '**1344**' is minimum output, so we insert 1344 in the table and M_1 x M_2 x(M_3 x M_4 x M_5)combination is taken out in output making.

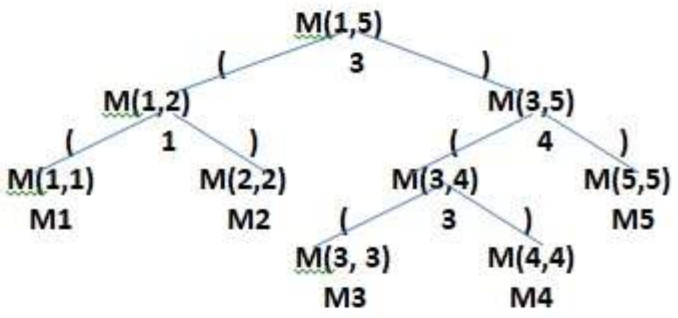
Final Output is:

	1	2	3	4	5
1	0	120	264	1080	
2		0	360	1320	1350
3			0	720	1140
4				0	1680
5					0

→

	1	2	3	4	5
1	0	120	264	1080	1344
2		0	360	1320	1350
3			0	720	1140
4				0	1680
5					0

[8 Marks]



((M1 M2)((M3 M4) M5))

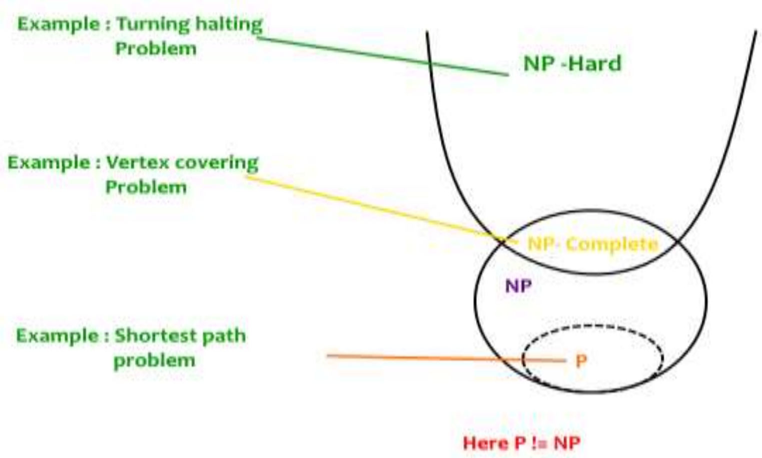
[2 Marks]

Sol (5-a): Write short notes on the following:

(i) **NP Completeness:** NP-Complete (NPC) problems are problems that are present in both the NP and NP-Hard classes. That is NP-Complete problems can be verified in polynomial time and any NP problem can be reduced to this problem in polynomial time.

A problem is in class NPC if it is in NP and is as hard as any problem in NP. A problem is said to be NP-hard if all problems in NP are polynomial time reducible to it, even though it may not be in NP itself.

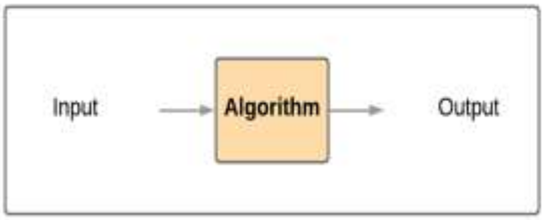
If a polynomial time algorithm exists for any of these types of problems, all problems in NP can be polynomial time solvable. These problems are called NP-complete. NPcompleteness is important for both theoretical and practical reasons.



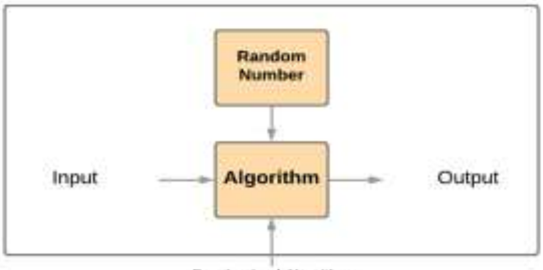
P is a set of problems that can be solved by a deterministic Turing machine in Polynomial-time.

NP is a set of decision problems that can be solved by a **N**on-deterministic Turing Machine in Polynomial-time. P is a subset of NP (any problem that can be solved by a deterministic machine in polynomial time can also be solved by a non-deterministic machine in polynomial time). **[5 Marks]**

(ii) **Randomization:** An algorithm that uses random numbers to decide what to do next anywhere in its logic is called Randomized Algorithm. For example, in Randomized Quick Sort, we use a random number to pick the next pivot (or we randomly shuffle the array). Typically, this randomness is used to reduce time complexity or space complexity in other standard algorithms.



Deterministic Algorithm Flowchart



Randomized Algorithm Flowchart

[2.5 Marks]

(iii) **Approximation:** Approximation algorithms are efficient algorithms that find approximate solutions to optimization problems (in particular NP-hard problems) with provable guarantees that the returned solution is optimal one. The goal of an approximation algorithm is to come as close as possible to the optimum value in a reasonable amount of time which is at the most polynomial time. Such algorithms are also called heuristic algorithm. • For the traveling salesperson problem, the optimization problem is to find the shortest cycle, and the approximation problem is to find a short cycle. • For the vertex cover problem, the optimization problem is to find the vertex cover with fewest vertices, and the approximation problem is to find the vertex cover with few vertices. There are many examples of approximation algorithms.
Some of them are:

i. Vertex-Cover Problem
ii. Set-Cover Problem
iii. Travelling Salesman Problem **[2.5 Marks]**

<u>Sol (5-b):</u> Given a string 'T' and pattern 'P' as follows:

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	c	a

P:	a	b	a	b	a	c	a

Let us execute the KMP Algorithm to find whether 'P' occurs in 'T.'
For 'p' the prefix function was computed previously and is as follows: **(2Marks)**

q	1	2	3	4	5	6	7
p	a	b	A	b	a	c	a
π	0	0	1	2	3	0	1

Initially: n = size of T = 15
m = size of P = 7

Step1: i=1, q=0

Comparing P [1] with T [1]

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b
P:	a	b	a	b	a	c	a								

P [1] does not match with T [1]. 'p' will be shifted one position to the right.

Step2: i = 2, q = 0

Comparing P [1] with T [2]

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b
P:		a	b	a	b	a	c	a							

P [1] matches T [2]. Since there is a match, p is not shifted.

Step 3: i = 3, q = 1

Comparing P [2] with T [3] P [2] doesn't match with T [3]

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b
P:		a	b	a	b	a	c	a							

Backtracking on p, Comparing P [1] and T [3]

Step4: i = 4, q = 0

Comparing P [1] with T [4] P [1] doesn't match with T [4]

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b
P:				a	b	a	b	a	c	a					

Step5: i = 5, q = 0

Comparing P [1] with T [5] P [1] match with T [5]

T:	b	a	c	b	a	b	a	b	a	b	a	c	a	a	b
P:					a	b	a	b	a	c	a				

Step6: i = 6, q = 1

Comparing P [2] with T [6]   P [2] matches with T [6]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step7: i = 7, q = 2

Comparing P [3] with T [7]   P [3] matches with T [7]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step8: i = 8, q =3

Comparing P [4] with T [8]   P [4] matches with T [8]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step9: i = 9, q = 4

Comparing P [5] with T [9]   P [5] matches with T [9]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step10: i = 10, q = 5

Comparing P [6] with T [10]   P [6] doesn't match with T [10]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Backtracking on p, Comparing P [4] with T [10] because after mismatch q = π [5] = 3

Step11: i = 11, q =4

Comparing P [5] with T [11]   P [5] match with T [11]

T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step12: i = 12, q = 5

Comparing P [6] with T [12]   P [6] matches with T [12]

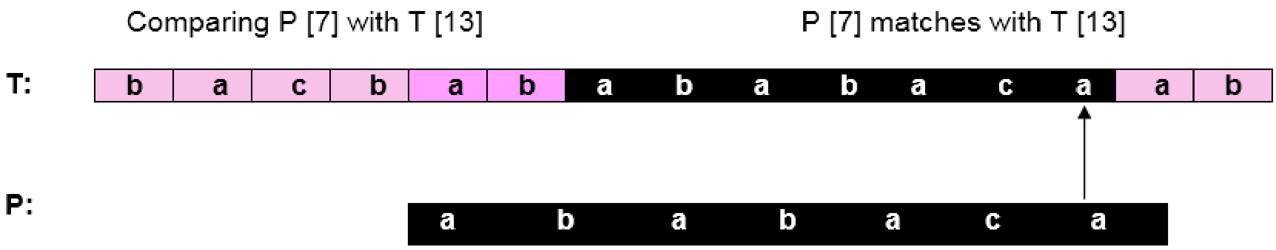
T:

b	a	c	b	a	b	a	b	a	b	a	c	a	a	b

P:

a	b	a	b	a	c	a

Step13: i = 3, q = 6



Pattern 'P' has been found in a string 'T.' The total number of shifts that took place for the match to be found is i-m = 13 - 7 = 6 shifts. **[8 Marks]**