23/5/24

Module – 02

Tree

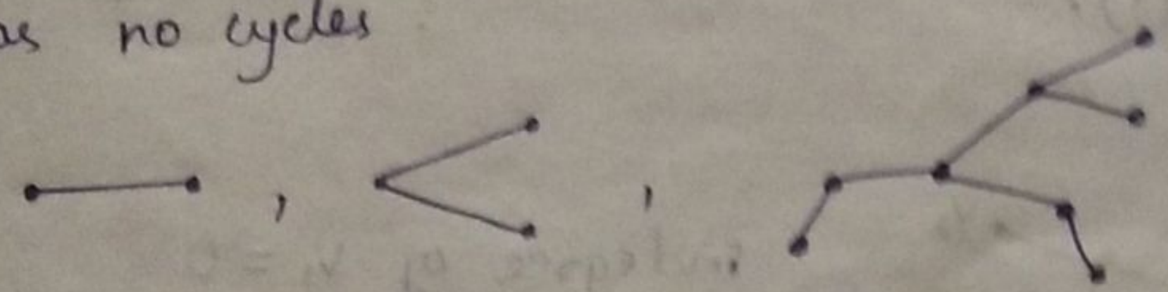
*** Definition of Tree:-**

A graph G is said to be a tree:-

(i) It should be connected

(ii) It has no cycles

Example:-



*** Leaf:-**

A pendant vertex of a tree is called leaf.

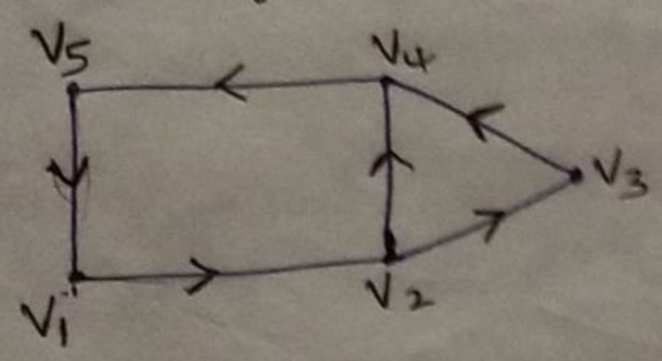
↓ a vertex with degree 1.

*** Directed graph:-**

A graph in which edges have direction is called directed graph.

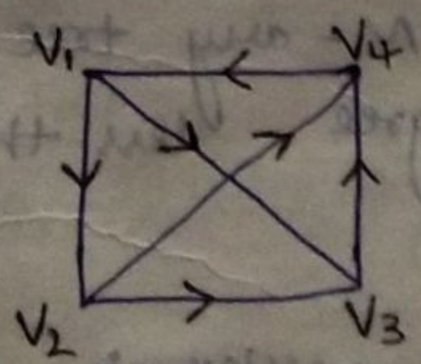
*** Indegree and outdegree of a graph:-**

Eg:- ①



Outdegree	Indegree
$V_1 \to 1$	$V_1 \to 1$
$V_2 \to 2$	$V_2 \to 1$
$V_3 \to 1$	$V_3 \to 1$
$V_4 \to 1$	$V_4 \to 2$
$V_5 \to 1$	$V_5 \to 1$

②



Outdegree	Indegree
$V_1 \to 2$	$V_1 \to 1$
$V_2 \to 2$	$V_2 \to 1$
$V_3 \to 1$	$V_3 \to 2$
$V_4 \to 1$	$V_4 \to 2$

Scanned By Camera Scanner

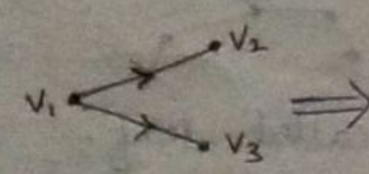
* **Rooted tree:-**

A directed tree 'T' is called Rooted tree if (i) 'T' contains unique vertex called root whose indegree is equal to zero.

(ii) The indegree of all other vertex of T are equal to one (1).

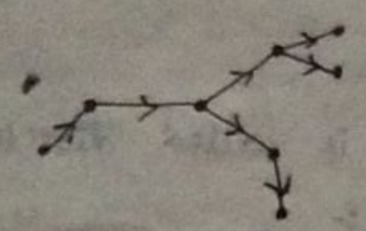
Example:-

①

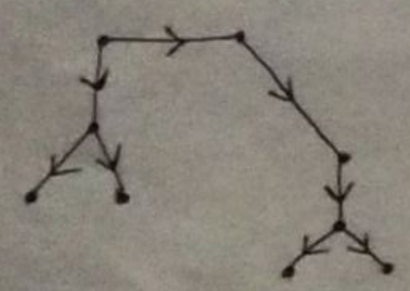


$\Longrightarrow$ indegree of $V_1 = 0$
indegree of $V_2 = 1$
indegree of $V_3 = 1$

②



③



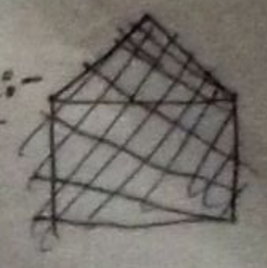
In a rooted tree a vertex whose outdegree is zero (0) is called leaf, A vertex which is not leaf is called internal vertex.

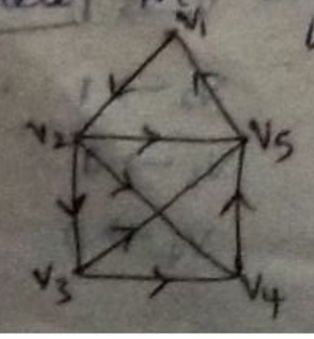
* **M-ary Tree:-** A Tree in a rooted tree has no more than m children

A Rooted Tree 'T' is called an M-ary tree if every internal vertex of 'T' is of outdegree less than or equal to M, then is every internal vertex of T has at most m children

Suppose outdegree of each internal vertex is exactly M then it is called complete M-ary tree.

Eg:- **M-ary tree:-**





	Outdegree	Indegree
V_1	1	1
V_2	3	1
V_3	2	1
V_4	1	2
V_5	1	3

* **Leaf:-** If outdegree is exactly 0 then it is called leaf.

* Other than the leaf is known as the internal vertex.

* **Binary Tree:-**

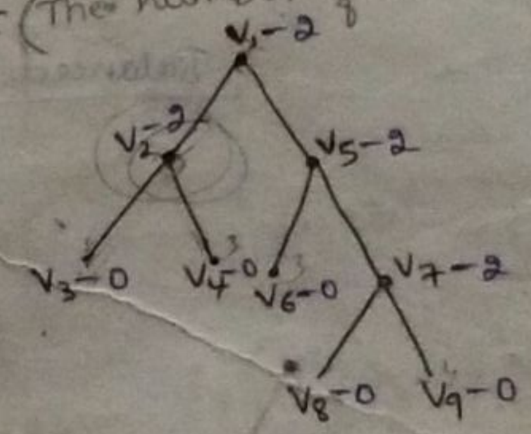
An M-ary tree for which $m = 2$ is called Binary tree.

i.e, A Rooted tree 'T' is called Binary tree if every vertex outdegree is less than or equal to 2.

Suppose each internal vertex outdegree is exactly 2 then it is called Complete Binary tree.

* **Height:-** (The number of subtrees)

Ex:-



* **Balanced tree:-**

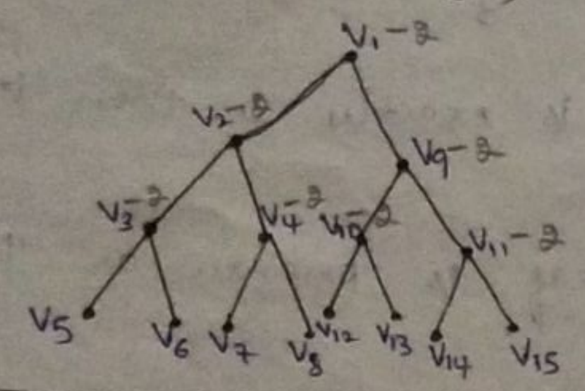
If 'T' is a rooted tree and 'h' is the largest level number achieved by a leaf of 'T' then T is said to have height h. A Rooted tree of height h is said to be balanced if the level number of every leaf is h or h-1.

Scanned By Camera Scanner

Full Binary tree:-

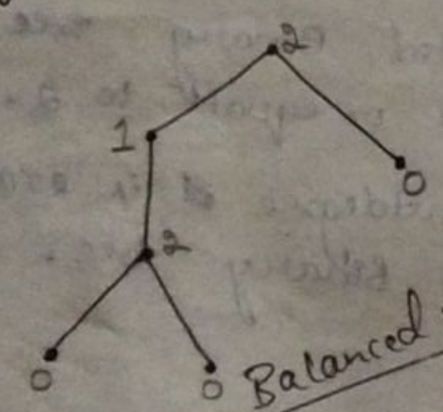
let 'T' be a complete binary tree of height H then T is said to be full binary tree if all the leaves in T are at same level.
(same height)

Eg:-



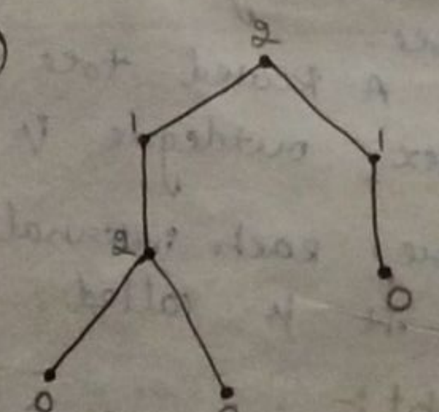
Which of the following is a Balanced tree and Full Binary - Tree :-

①



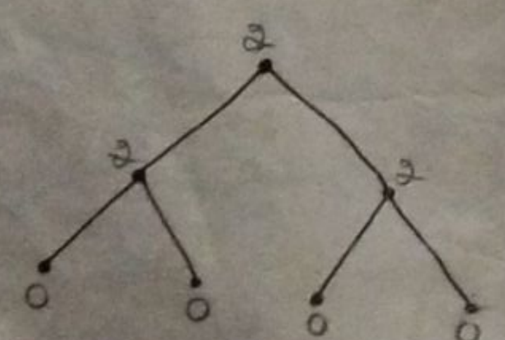
Balanced tree

②



Balanced Tree

③



Full Binary Tree

Spanning Tree:-

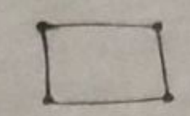
Let 'G' be a ~~graph~~ connected graph. A subgraph T of G is called spanning Tree if it satisfies following conditions:-

(i) It should be a tree — It should be connected, No cycles.

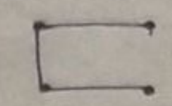
(ii) It should contain all the vertices of G.

Find all the Spanning trees of the graph:-

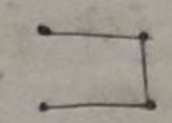
①



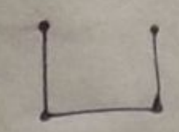
(i)



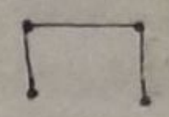
(iii)



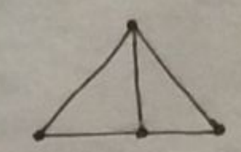
(ii)



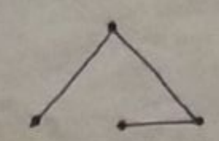
(iv)



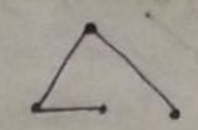
②



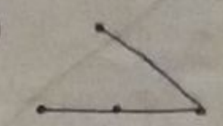
(i)



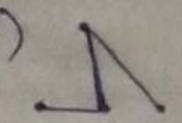
(iii)



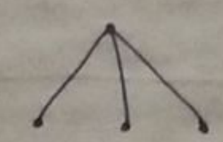
(v)



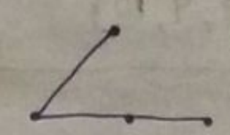
(vii)



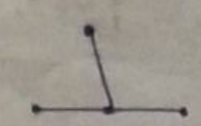
(ii)



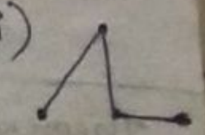
(iv)



(vi)



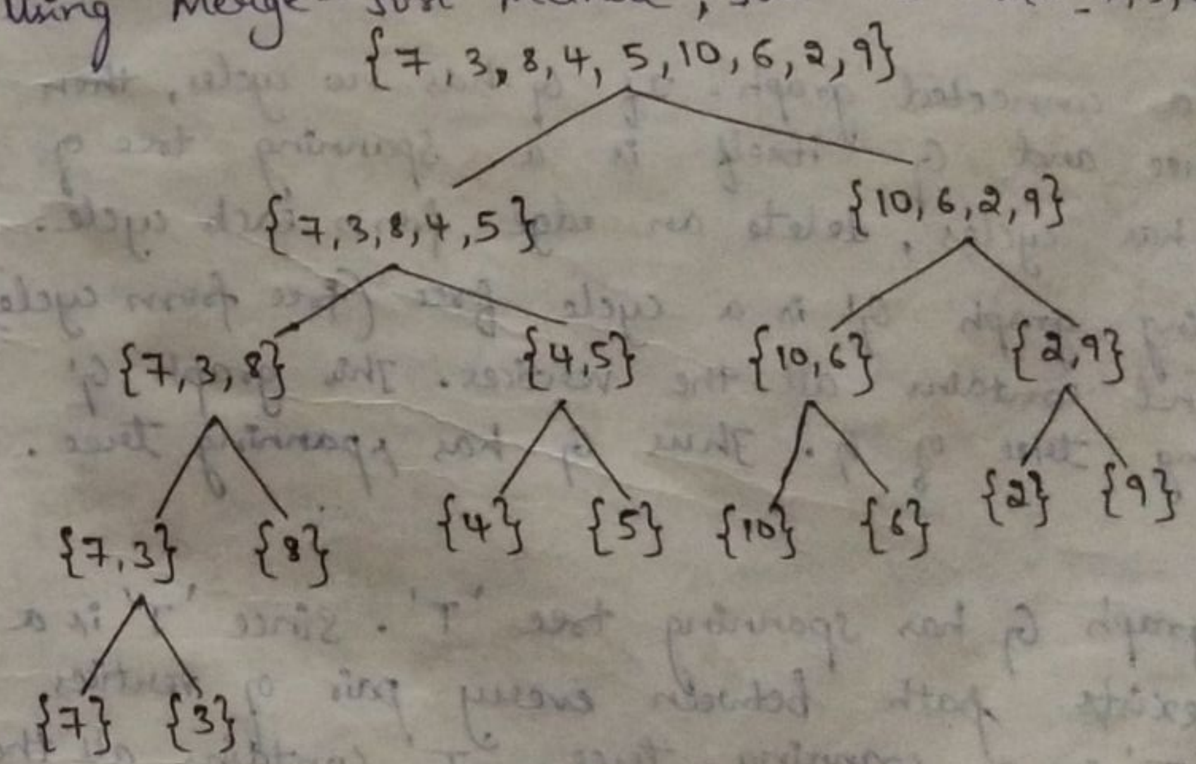
(viii)



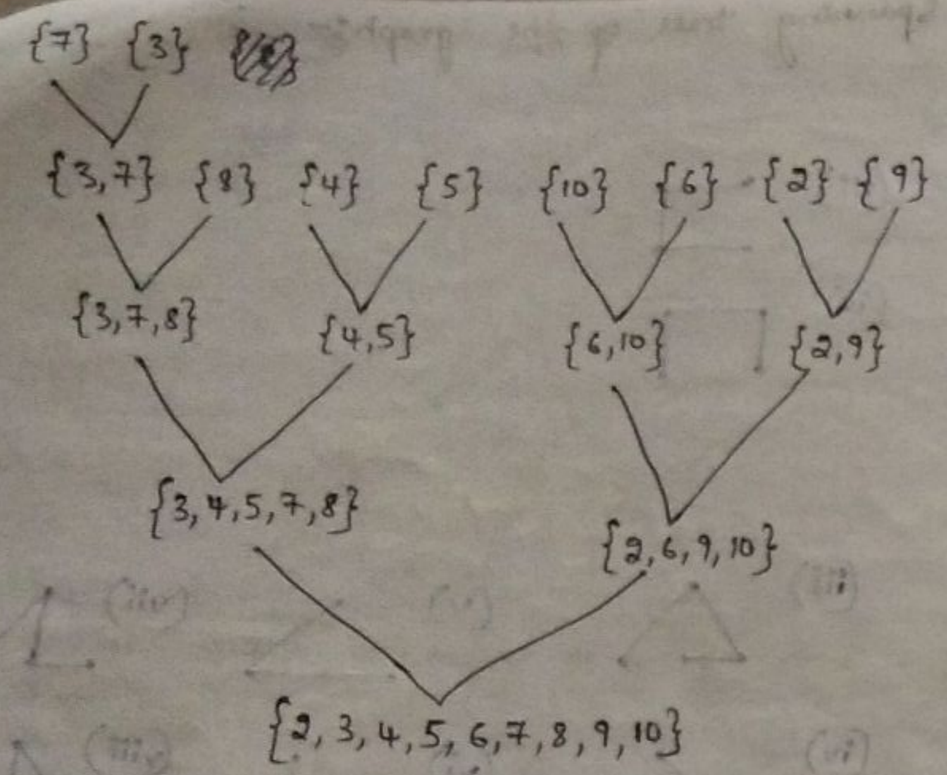
* Sorting:-

splitting & merging process is done by using binary search tree

Using Merge-Sort Method, sort the list 7, 3, 8, 4, 5, 10, 6, 2, 9.



Scanned By Camera Scanner



Theorem 1:-

① A graph 'G' is connected if and only if it has a spanning tree.

Let G be a connected graph. If G has no cycles, then G is a tree and G itself is a spanning tree of G. If G has cycles, delete an edge from each cycle. The resulting graph G' is a cycle free (free from cycle), connected and contains all the vertices. This graph G' is a spanning tree of G. Thus G has spanning tree.

Conversely,

Suppose a graph G has spanning tree 'T'. Since 'T' is a tree there exists path between every pair of vertices in 'T'. Since 'T' is a spanning tree 'T' contain all the vertices of G. Therefore there is a path between every pair of vertices in G. Hence G is a connected graph.

Theorem – 2:-

② A connected graph is a tree if and only if it is minimally connected. disconnected.

Suppose G is a connected graph which is not a tree. Then G contains cycle C. The Removal of any one edge 'e' from this cycle will not make the graph disconnected. Therefore G is not minimally connected. Thus if a connected graph is ~~not~~ a tree then it is not minimally connected. This is equivalent to saying that if a connected graph is minimally connected then it is a tree.

Conversely,

Suppose G is a connected graph which is not minimally connected. Then there exist an edge 'e' in G such that G−e is connected. Therefore e must be in some cycle in G. This implies G is not a tree. Thus if a connected graph is not minimally connected then it is not a tree. This is equivalent to saying that if a connected graph is a tree then it is minimally connected.

Theorem – 3:-

③ In a tree, there is one and only one path between every pair of vertices.

Let 'T' be a tree. Then T is a connected simple graph. Since T is connected there must be atleast one path between every pair of vertices. If there are two paths between every pair of vertices of T then the union of the path will become a graph and G cannot be a tree. Thus between every pair of vertices in a tree there must exists one and only one path.

④ Theorem – 4:-

If in a simple ~~the~~ graph G, there is one and only one path between every pair of vertices then G is a tree.

Since there is a path between every pair of vertices in G then G is connected. Since there is only one path

Scanned By Camera Scanner

between every pair of vertices then G cannot have cycle. Because if there is a cycle then there exits two paths between ~~every pair~~ two vertices on the cycle. Thus G is a connected graph containing no cycle. This means that G is a tree.

Prefix Code:-

Binary Sequence:- A sequence consisting of only 0 and 1 is called Binary Sequence.

Eg:- 01, 0, 1, 101, 00101, 1111, 1001

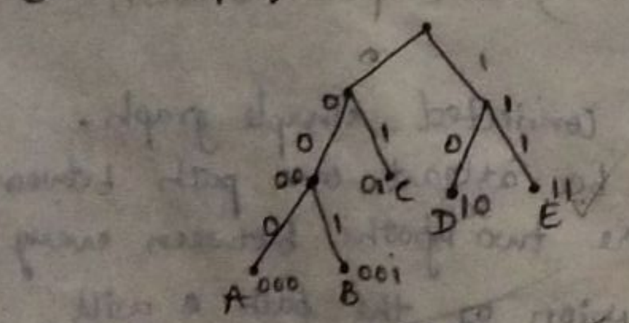
Prefix Code:-

Let P be the set of binary sequences that represents a set of symbols. Then P is called prefix code if no sequence in P is the prefix for any other sequence in P.

Example:- $P_1 = \{10, 0, 1101, 111, 1100\}$

$P_2 = \{01, 0, 101, 1101\}$ —— not a prefix code because 0 is the prefix for 01.

① Find prefix code for ABCDE:-



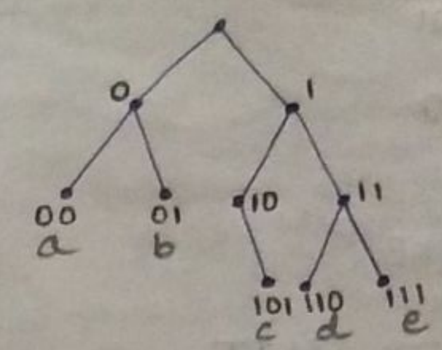
Take left side as 0
Take right side as 1.

Ans:-

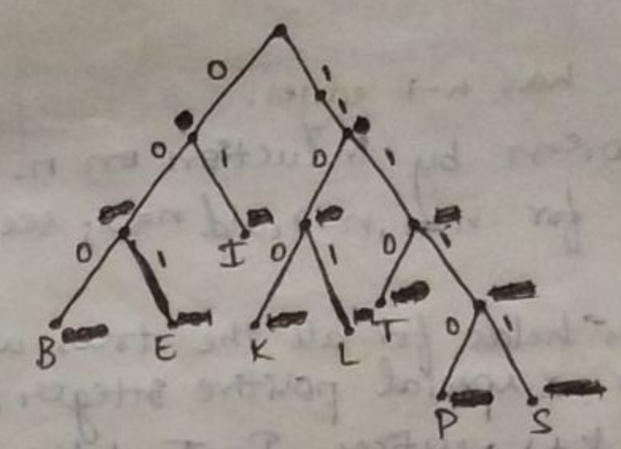
A → 000
B → 001
C → 01
D → 10
E → 11

② A code for {a, b, c, d, e} is given by a = 00, b = 01, c = 101, d = 110, e = 111

Ans:-



③ Find prefix code for letters B, E, I, K, L, T, P, S.



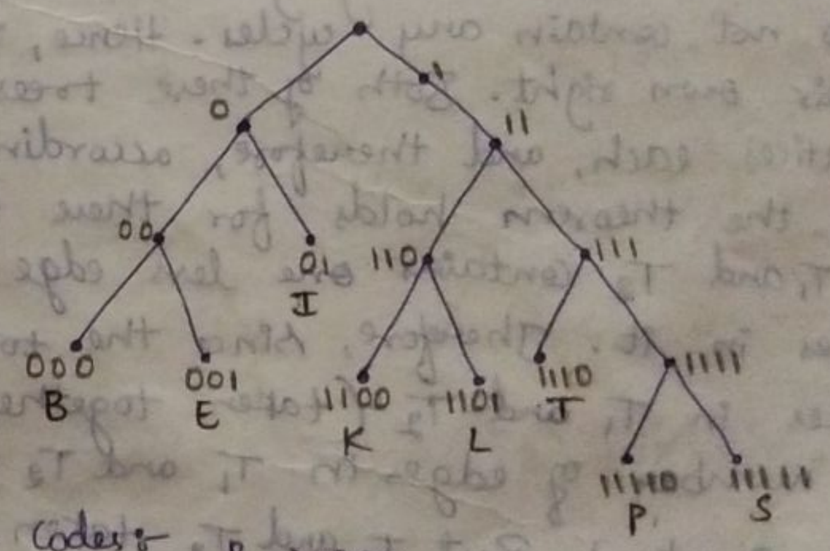
① Also find codes for PIPE and BEST

② Decode the strings

(i) 000011100001

(ii) 1111111110110101110

Ans:-



Prefix Codes:-

B → 000
E → 001
I → 01
K → 1100
L → 1101
T → 1110
P → 11110
S → 11111

Scanned By Camera Scanner

(i) PIPE — 1111001111 0001

BEST — 0000011111110

(ii)

$\underset{B}{000}\underset{I}{0111}\underset{K}{00}\underset{E}{001}$ — BIKE

$\underset{S}{111}\underset{P}{111111}\underset{L}{0110}\underset{I}{1011}\underset{T}{110}$ — SPLIT

Theorem :-

(5) A Tree with n vertices has n-1 edges.

Proof :- We prove the theorem by induction on n.

The theorem is obvious for n=1, n=2 and n=3; see the trees in Figure 3.1.

Assume that the theorem holds for all the trees with n vertices where $n \le k$, for a special positive integer k.

Consider a tree T with k+1 vertices. In T, let e be an edge with end vertices u and v. Since T is a tree, it has no cycles and therefore there exists no other edge or path between u and v. Hence, deletion of e from T will disconnect the graph and T-e consists of exactly two components, say T_1 and T_2. Since T does not contain any cycle, the components T_1 and T_2 too do not contain any cycles. Hence, T_1 and T_2 are trees in their own right. Both of these trees have less than k+1 vertices each, and therefore, according to the assumption made, the theorem holds for these trees; that is, each of T_1 and T_2 contains one less edge than the number of vertices in it. Therefore, since the total number of vertices in T_1 and T_2 (taken together) is k+1, the total number of edges in T_1 and T_2 (taken together) is $(k+1)-2 = k-1$. But T_1 and T_2 taken together is T-e. Thus, T-e contains k-1 edges. Consequently, T has exactly k edges.

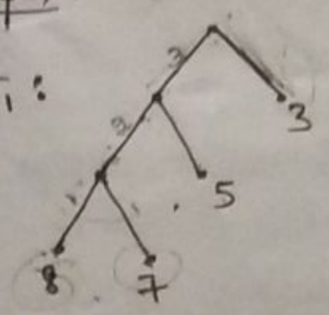
Weighted tree :-

Consider a set of n positive integers $w_1, w_2, \ldots, w_n$ where $w_1 \le w_2 \le w_3$. Suppose we assign these integers to n leaves of a complete Binary tree 'T' in any one-to-one manner. The resulting tree is called complete, weighted, Binary tree with $w_1, w_2, \ldots, w_n$ as weights. If $l(w_i)$ is the level number of the leaf of 'T' on which the weight w_i assign then weight of tree is denoted by $w(T)$.

$$\boxed{w(T) = \sum_{i=1}^{n} w_i * l(w_i)}$$

Examples :-

(1) T_1 :



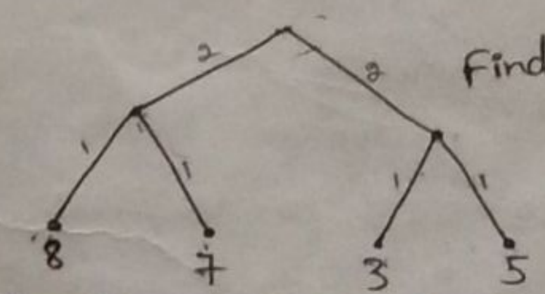
Find weight of a tree.

$= (8\times3)+(7\times3)+(5\times2)+(3\times1)$

$= 24+21+10+3$

$\boxed{w(T_1) = 58}$

(2) T_2 :



Find weight of a tree.

$= (8\times2)+(7\times2)+(3\times2)+(5\times2)$

$= 16+14+6+10$

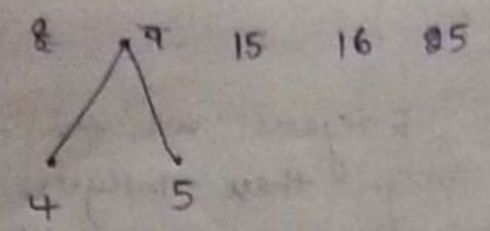
$= 46$

#(1) Construct optimal tree for the weights (4, 15, 25, 5, 8, 16) and then find weight of the tree.

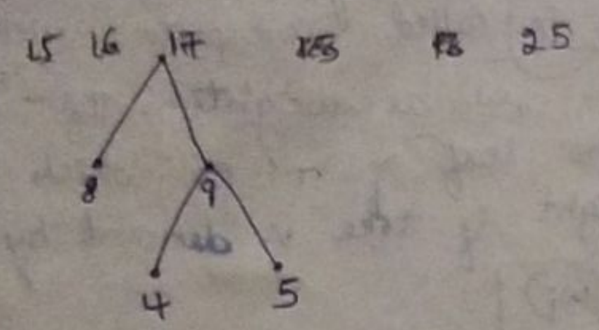
Ans :- {4, 15, 25, 5, 8, 16}

{4, 5, 8, 15, 16, 25}

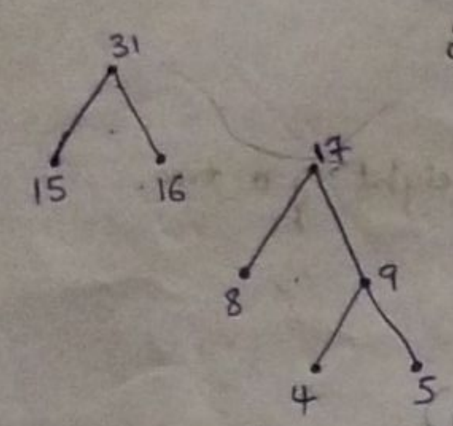
I :-



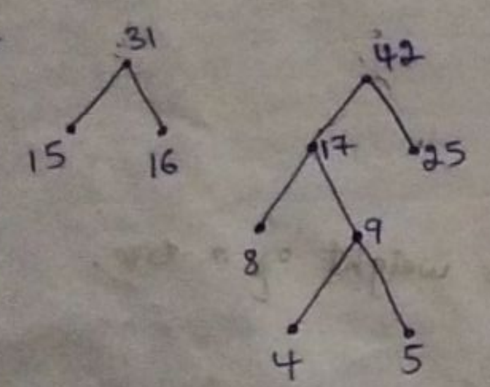
II :-



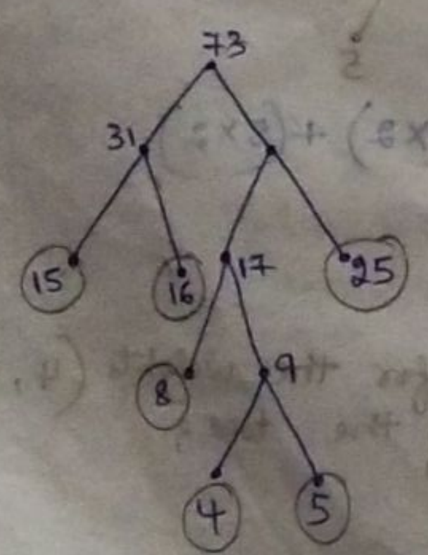
III :-



IV :-



V :-



weight of a tree ⇒ $T = (4\times4)+(5\times4)+(8\times3)+(15\times2)+(16\times2)+(25\times2)$

$\boxed{w(T) = 172}$

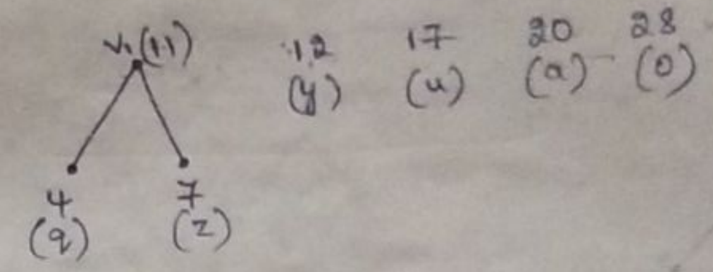
② Construct the ~~prefix~~ ~~code~~ optimal prefix codes for symbols $\{a, o, q, u, y, z\}$ that occur with frequencies 20, 28, 4, 17, 12, 7.

Ans :-

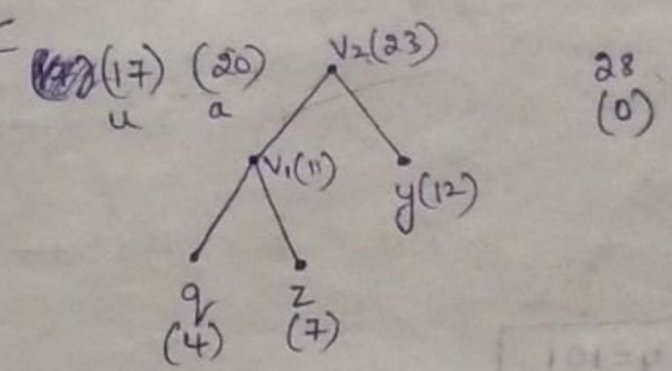
a	o	q	u	y	z
20	28	4	17	12	7

4 (q) 7 (z) 12 (y) 17 (u) 20 (a) 28 (o)

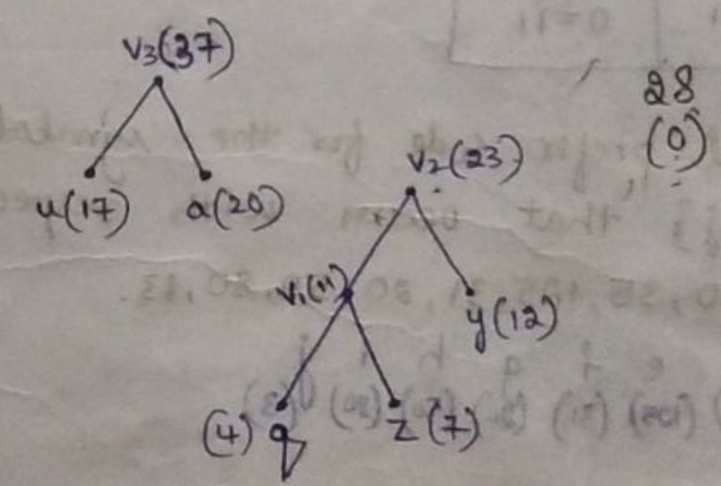
I :-



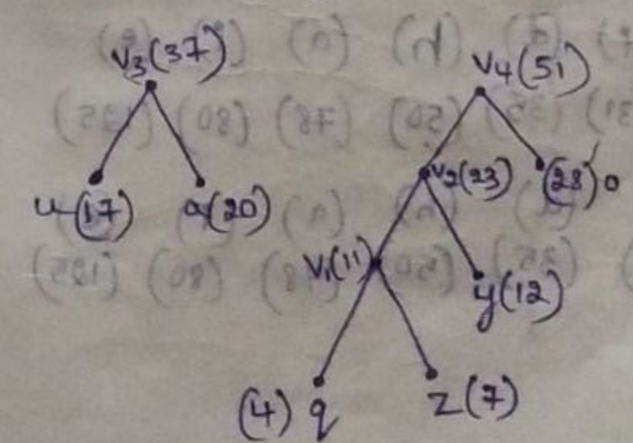
II :-



III :-

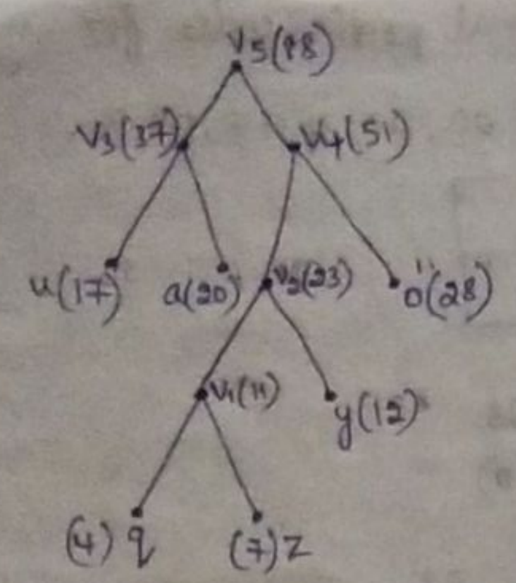


IV :-

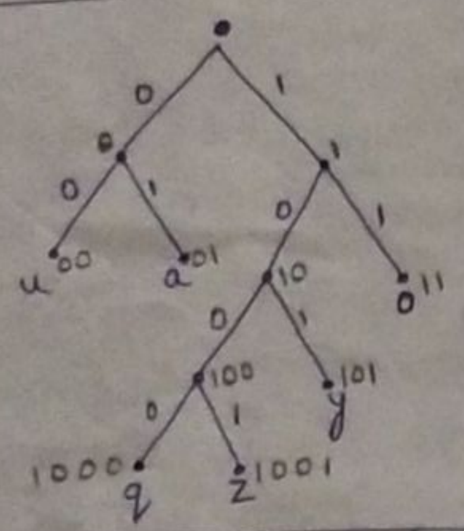


Scanned By Camera Scanner

IV :-



Prefix codes :-



u = 00	q = 1000	y = 101
a = 01	z = 1001	o = 11

③ Construct an optimal prefix code for the symbols {a, b, c, d, e, f, g, h, i, j} that occur with respective frequencies 78, 16, 30, 35, 125, 31, 20, 50, 80, 3.

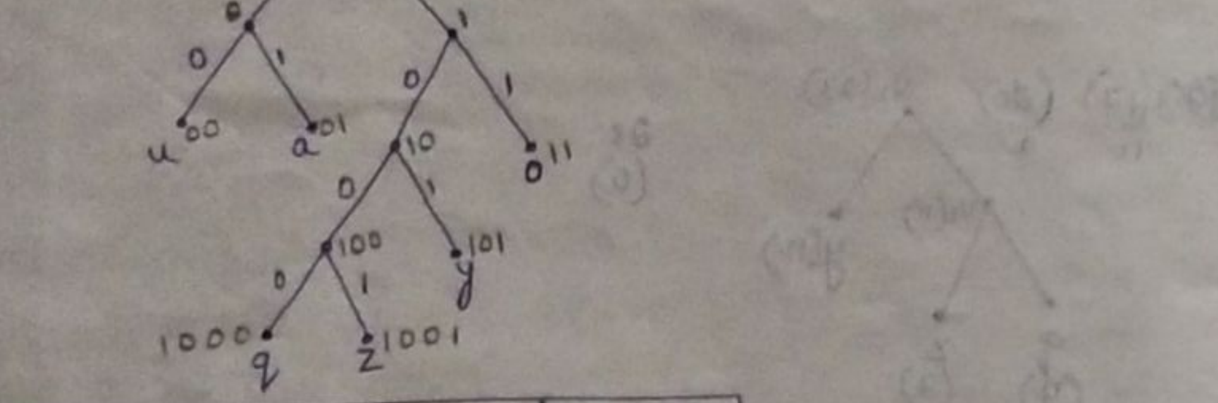
Ans :-

a	b	c	d	e	f	g	h	i	j
(78)	(16)	(30)	(35)	(125)	(31)	(20)	(50)	(80)	(3)

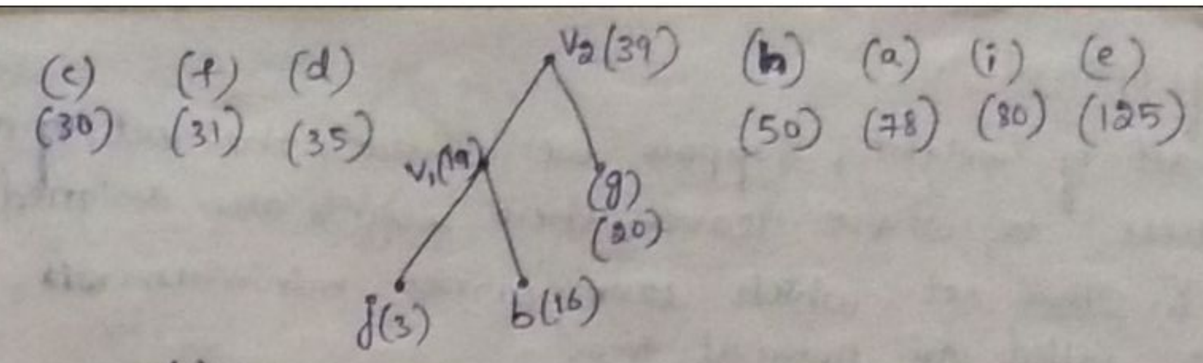
Ans

(j)	(b)	(g)	(c)	(f)	(d)	(h)	(a)	(i)	(e)
(3)	(16)	(20)	(30)	(31)	(35)	(50)	(78)	(80)	(125)

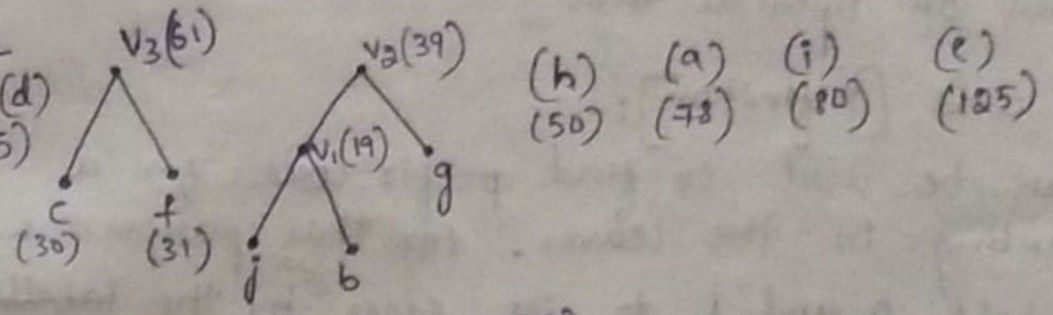
I :-



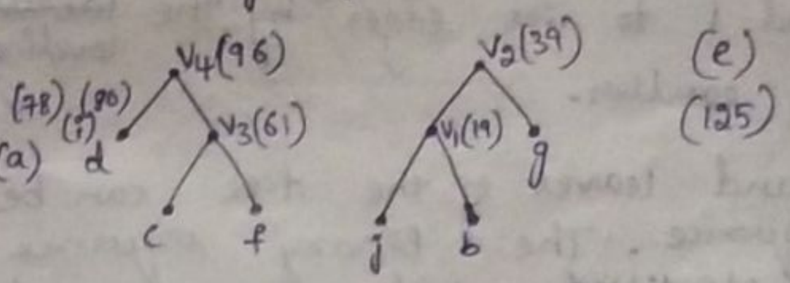
II :-



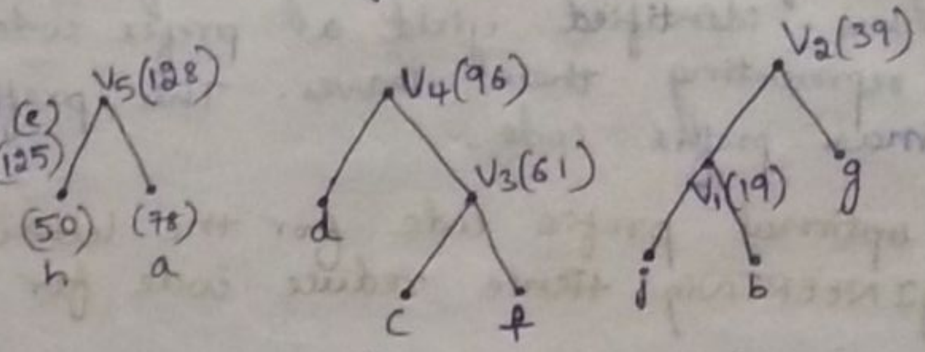
III :-



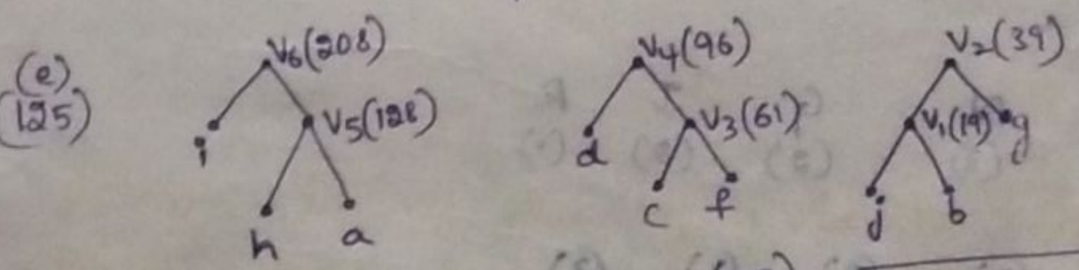
IV :-



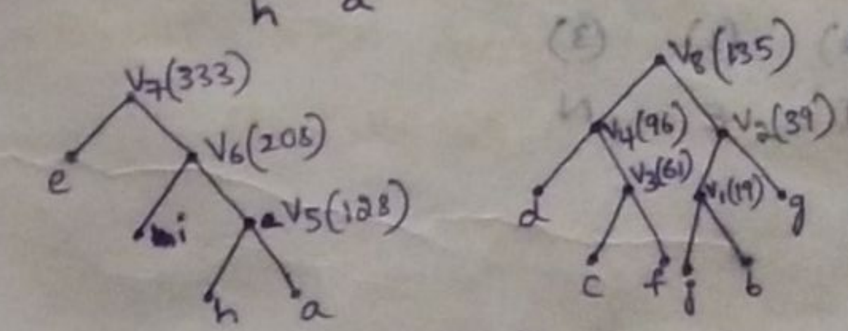
V :-



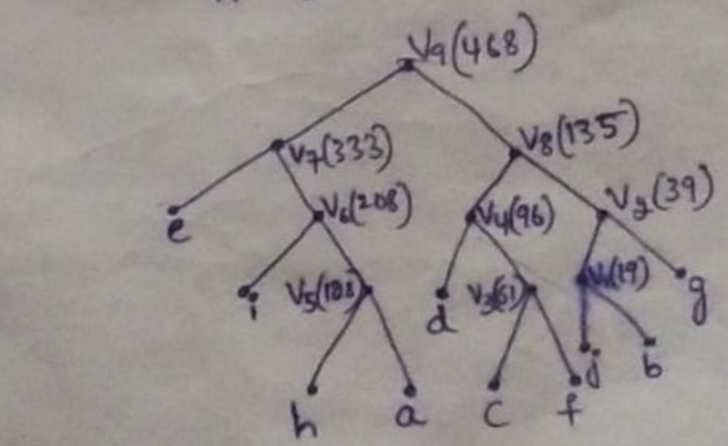
VI :-

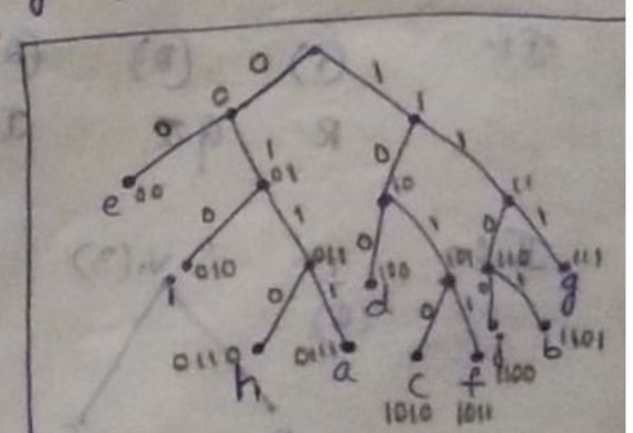


VII :-



VIII :-





e → 00
i → 010
h → 0110
a → 0111
d → 100
c → 1010
f → 1011
j → 1100
b → 1101
g → 111

Scanned By Camera Scanner

* Definition of optimal tree:-

Given a set of weights, suppose we consider the set of all binary trees to whose leaves these weights are assigned. A tree in this set which carries minimum weight is called an optimal tree.

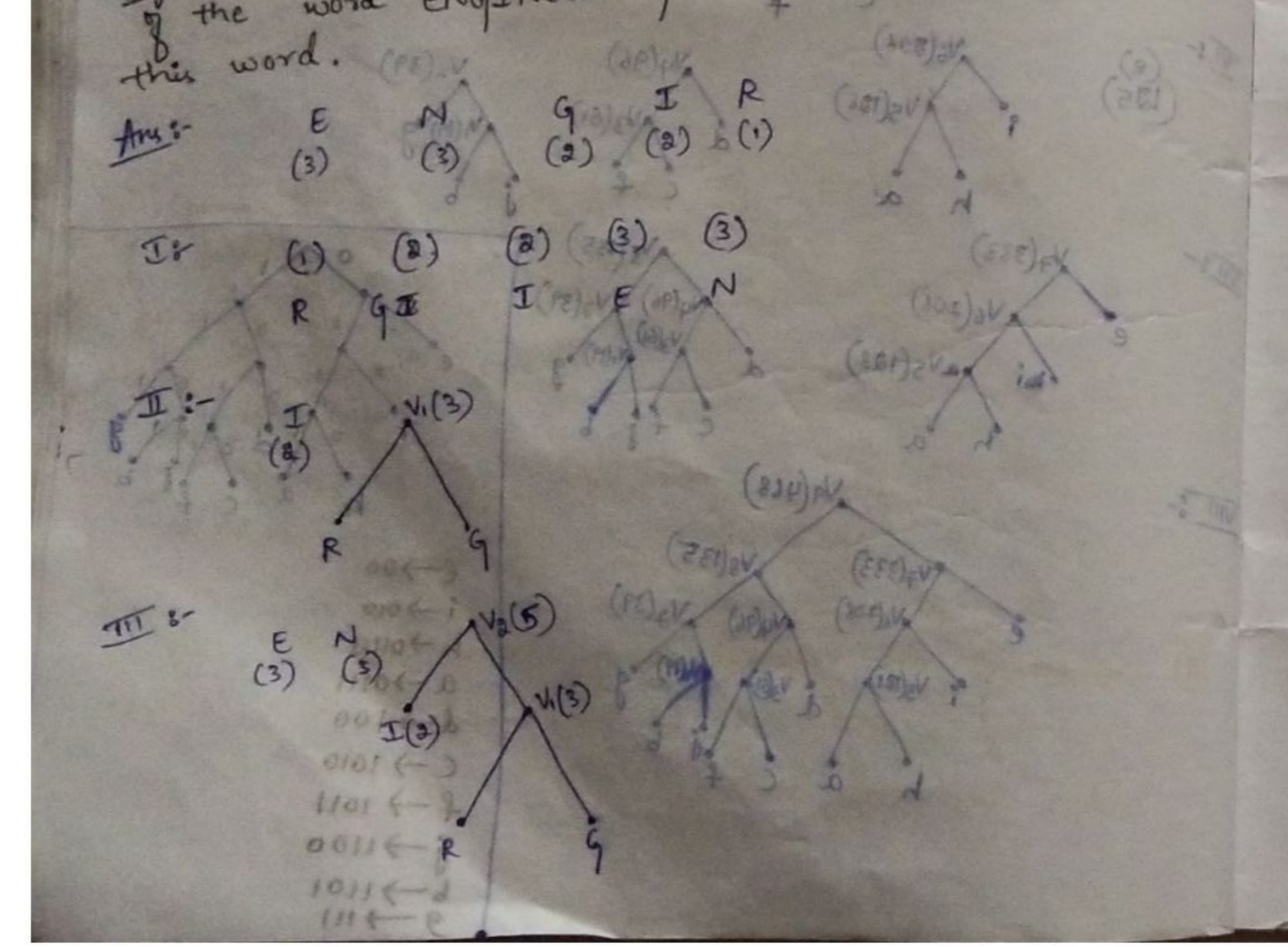
* Optimal prefix code [Definition]:-

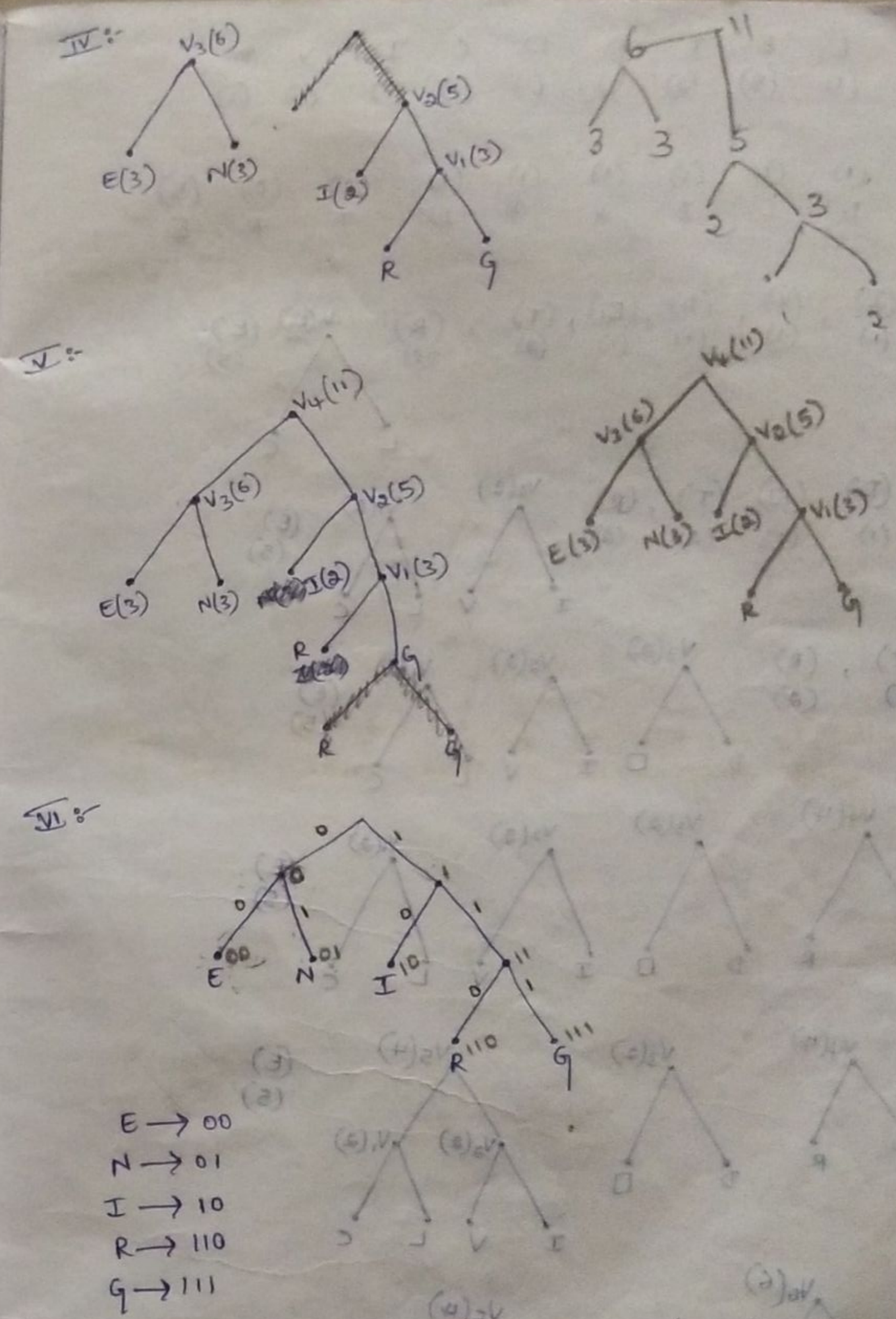
Optimal tree can be used to find prefix code for a symbols representing in the leaves. For this purpose we first label symbols 0 and 1 to its edges by the ~~labelling~~ leveling procedure as indicated earlier.

Then all the vertices and leaves of the tree can be identified by binary sequence. The Binary sequence to which leaves are identified yield a prefix code for the symbols representing these leaves. This prefix code is called optimal prefix code.

Eg:- ① Construct an optimal prefix code for the letters of the word ENGINEERING. Hence reduce code for this word.

Ans:-





E → 00
N → 01
I → 10
R → 110
G → 111

② Construct an optimal prefix code for the letters of the word LETTER RECEIVED. Hence reduce code for this word.

Scanned By Camera Scanner

Ans :-

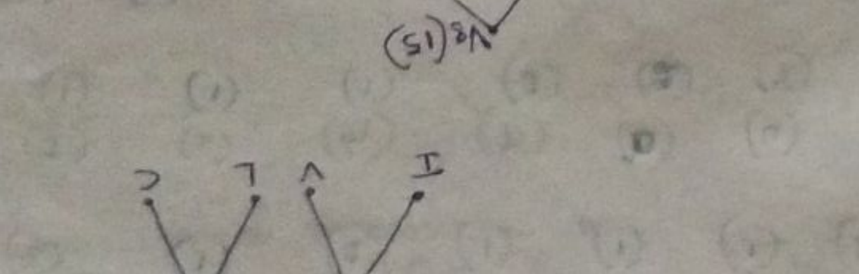
L	E	T	R	□	C	I	V	D
(1)	(5)	(3)	(3)	(1)	(1)	(1)	(1)	(1)

I :-

L	C	I	V	D	□	T	R	E
(1)	(1)	(1)	(1)	(1)	(1)	(3)	(3)	(5)

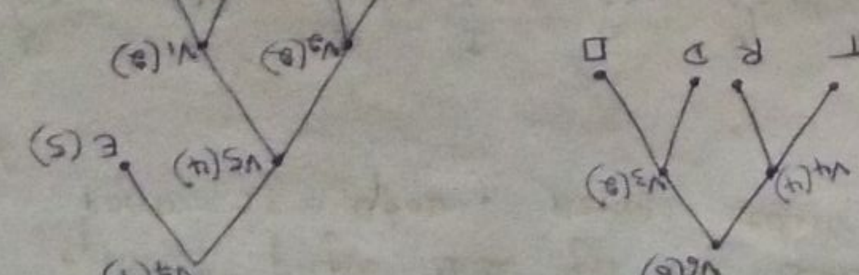
II :-

(I) (1), (V) (1), (D) (1), (□) (1), (T) (3), (R) (3), $v_1(8)$ (5), (E) (5)



III :-

(D) (1), (□) (1), (T) (2), (R) (3), $v_2(8)$ (5), $v_1(8)$ (5), (E) (5)



IV :-

(T) (3), (R) (3), $v_3(8)$ (5), $v_2(8)$ (5), $v_1(8)$ (5), (E) (5)

V :-

$v_4(4)$ $v_3(8)$ $v_5(4)$ (E) (5) $v_3(8)$ $v_1(8)$

VI :-

$v_6(6)$ $v_4(4)$ $v_3(8)$ $v_5(4)$ (E) (5) $v_2(8)$ $v_1(8)$

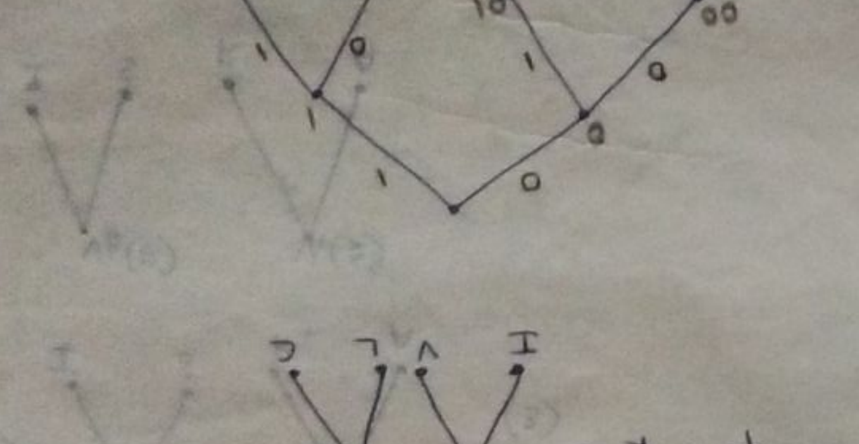
VII :-

$v_6(6)$ $v_4(4)$ $v_3(8)$ $v_7(9)$ $v_5(4)$ E(5) $v_2(8)$ $v_1(8)$

VIII :-

$v_8(15)$ $v_7(9)$ $v_6(6)$ $v_4(4)$ $v_5(4)$ E(5) $v_3(8)$ $v_2(8)$ $v_1(8)$

IX :-



L → 1010
E → 11
V → 1001
T → 000
D → 010
R → 001
□ → 011
C → 1011
I → 1000

Scanned By Camera Scanner

③ Construct an optimal prefix code for the letters of the word ROAD IS GOOD. Hence reduce code for this word.

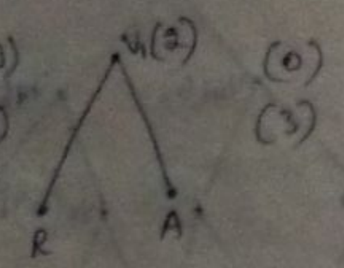
Ans:-

R	O	A	D	I	S	G	□
(1)	(3)	(1)	(2)	(1)	(1)	(1)	(2)

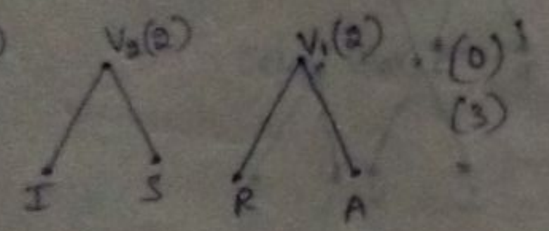
I:-

(R)	(A)	(I)	(S)	(G)	(D)	(□)	(O)
(1)	(1)	(1)	(1)	(1)	(2)	(2)	(3)

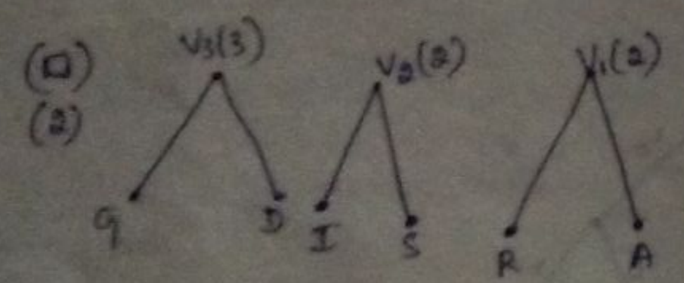
II:-



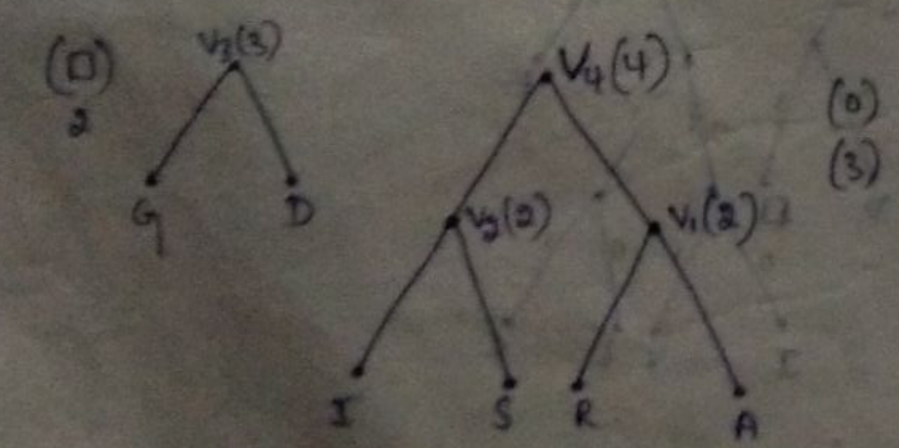
III:-



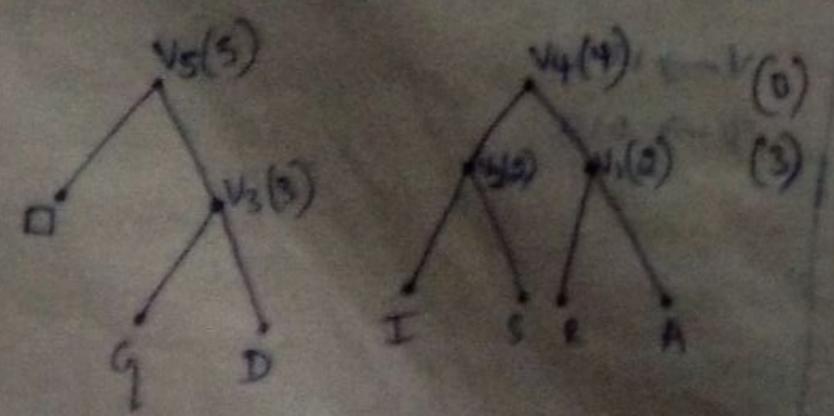
IV:-



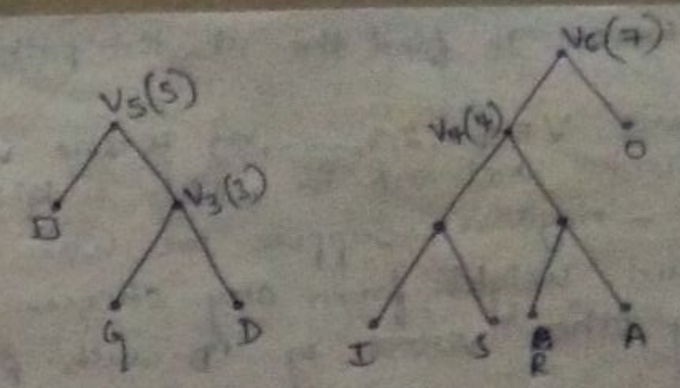
V:-



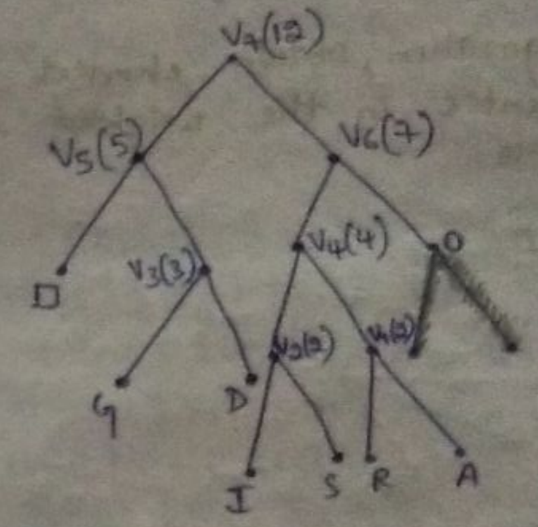
VI:-



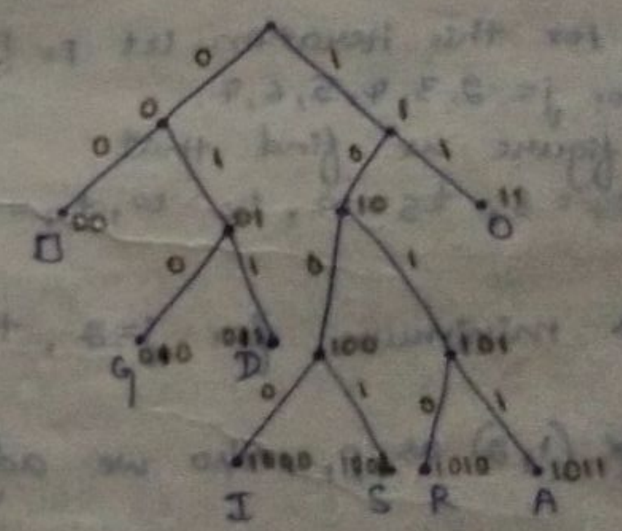
VII:-



VIII:-



IX:-



R ⟶ 1010

A ⟶ 1011

I ⟶ 1000

S ⟶ 1001

G ⟶ 010

D ⟶ 011

□ ⟶ 00

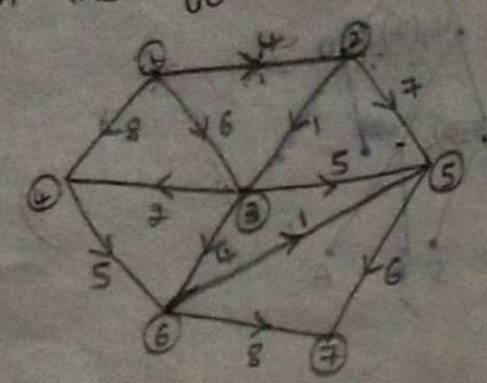
O ⟶ 11

Scanned By Camera Scanner

Dijkstra's Algorithm:- To find the shortest path b/w the vertices.

Let $D = D(V,E)$ where $V = \{1,2,3,\ldots,n\}$ is the vertex set, be a weighted, directed network in which weight of every directed edge is non-negative. Suppose we wish to find a path with minimum weight from any choosen vertex say vertex '1' to any other vertex of D such path is called shortest path.

Eg:- ① Using Dijkstra's Algorithm, obtain shortest path from vertex '1' to other vertex in the weighted directed network shown in the figure.



Ans:- **First Iteration:-** For this iteration let $P = \{1\}$ and $t_j = q_{1j}$ for $j = 2,3,4,5,6,7$.

By examining the figure we find that $t_2 = 4$, $t_3 = 6$, $t_4 = 8$, $t_5 = \infty$, $t_6 = \infty$, $t_7 = \infty$

Step-01:-

We ~~know~~ note that t_j is minimum for $j=2$, the minimum being $t_2 = 4$

$\therefore$ we label the edge $(1,2)$ as P_1. Also we adjoin 2 to P so that $P = \{1,2\}$

Step-02:-

We have $P = \{1,2\}$ and $t_2 = 4$ we choose $k=2$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 3,4,5,6,7$

new $t_3 = \min\{t_3, t_2 + q_{23}\} = \min\{6, 4+1\} = \min\{6,5\} = 5$

new $t_4 = \min\{t_4, t_2 + q_{24}\} = \min\{8, 4+\infty\} = 8$

new $t_5 = \min\{t_5, t_2 + q_{25}\} = \min\{\infty, 4+7\} = \min\{\infty, 11\} = 11$

new $t_6 = \min\{t_6, t_2 + q_{26}\} = \min\{\infty, 4+\infty\} = \infty$

new $t_7 = \min\{t_7, t_2 + q_{27}\} = \min\{\infty, 4+\infty\} = \infty$

Second Iteration:-

For this iteration $P = \{1,2\}$ and $t_2 = 4$. Also we have, $t_3 = 5$, $t_4 = 8$, $t_5 = 11$, $t_6 = \infty$, $t_7 = \infty$.

Step-01:-

We ~~know~~ note that among new t_j the minimum t_j occurs for $j=3$ the minimum being $t_3 = 5$. We observe that among the arcs (edge) from vertex in P to the vertex 3. The arc $(2,3)$ has least weight.

$\therefore$ Label the arc $(2,3)$ as P_2. Also we adjoin 3 to P, so that $P = \{1,2,3\}$

Step-02:-

We have $P = \{1,2,3\}$, $t_2 = 4$, $t_3 = 5$ we choose $k=3$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 4,5,6,7$

new $t_4 = \min\{t_4, t_3 + q_{34}\} = \min\{8, 5+2\} = \min\{8,7\} = 7$

new $t_5 = \min\{t_5, t_3 + q_{35}\} = \min\{11, 5+5\} = 10$

new $t_6 = \min\{t_6, t_3 + q_{36}\} = \min\{\infty, 5+4\} = 9$

new $t_7 = \min\{t_7, t_3 + q_{37}\} = \min\{\infty, 5+\infty\} = \infty$

Third Iteration:-

For this Iteration $P = \{1,2,3\}$, $t_2 = 4$, $t_3 = 5$. Also, $t_4 = 7$, $t_5 = 10$, $t_6 = 9$, $t_7 = \infty$.

Step-01:-

We note that among new t_j the minimum t_j occurs for $j=4$ the minimum being $t_4 = 7$. We observe that among the arcs from vertex in P to the vertex 4. The arc $(3,4)$ has least weight.

$\therefore$ label the arc $(3,4)$ as P_3. Also we adjoin 4 to P, so that $P = \{1,2,3,4\}$.

Step-02:-

We have $P = \{1,2,3,4\}$, $t_2 = 4$, $t_3 = 5$, $t_4 = 7$ we choose $k=4$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 5,6,7$

new $t_5 = \min\{t_5, t_4 + q_{45}\} = \min\{10, 7+\infty\} = 10$

new $t_6 = \min\{t_6, t_4 + q_{46}\} = \min\{9, 7+5\} = \{9, 12\} = 9$

new $t_7 = \min\{t_7, t_4 + q_{47}\} = \min\{\infty, 7+\infty\} = \infty$

Fourth Iteration:-

For this Iteration $P = \{1,2,3,4\}$, $t_2 = 4$, $t_3 = 5$, $t_4 = 7$. Also $t_5 = 10$, $t_6 = 9$, $t_7 = \infty$

Scanned By Camera Scanner

Step-01:-

We note that among new t_j the minimum t_j occurs for $j=6$ the minimum being $t_6=9$. We observe that among the arcs (edge) from the vertex in P to the vertex 6. The arc (3,6) has least weight.

∴ label arc (3,6) as P_3. Also we adjoin 6 to P,

So that $P=\{1,2,3,4,6\}$.

Step-02:-

We have $P=\{1,2,3,4,6\}$, $t_2=4$, $t_3=5$, $t_4=7$, $t_6=9$ we choose

$k=6$

new $t_j=\min\{t_j, t_k+q_{kj}\}$ $j=5,7$

new $t_5=\min\{t_5, t_6+q_{65}\}=\min\{10, 9+1\}=10$

new $t_7=\min\{t_7, t_6+q_{67}\}=\min\{\infty, 9+8\}=17$

Fifth Iteration:-

For this iteration $P=\{1,2,3,4,6\}$, $t_2=4$, $t_3=5$, $t_4=7$, $t_6=9$.

Also $t_5=10$, $t_7=17$.

Step-01:-

We note that among new t_j the minimum t_j occurs for $j=5$ the minimum being $t_5=10$. We observe that among the arcs (edge) from the vertex in P to the vertex 5. The arc (6,5) has least weight.

∴ label arc (6,5) as P_4. Also we adjoin 5 to P, so that $P=\{1,2,3,4,6,5\}$

Step-02:-

We have $P=\{1,2,3,4,6,5\}$, $t_2=4$, $t_3=5$, $t_4=7$, $t_6=9$, $t_5=10$

we choose

$k=5$

new $t_j=\min\{t_j, t_k+q_{kj}\}$ $j=7$

new $t_7=\min\{t_7, t_5+q_{57}\}=\min\{17, 10+6\}=16$

Final Iteration:-

$P=\{1,2,3,4,6,5,7\}$, $t_2=4$, $t_3=5$, $t_4=7$, $t_6=9$, $t_5=10$, $t_7=16$

① $1 \to 2 = 4$

② $1 \to 2 \to 3 = 5$

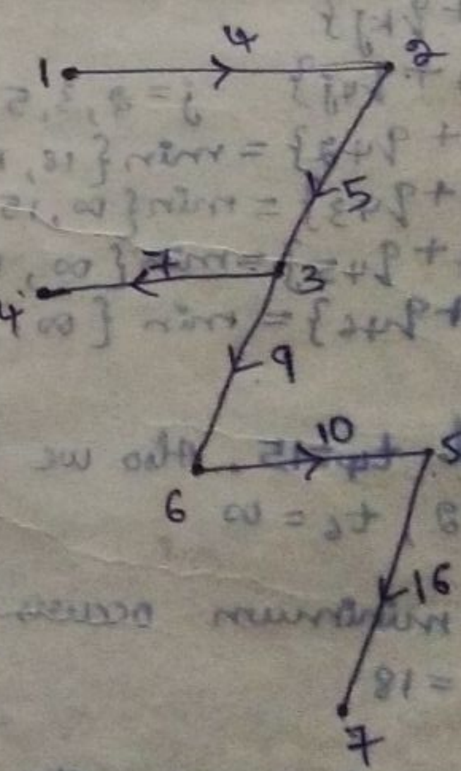
③ $1 \to 2 \to 3 \to 4 = 7$

④ $1 \to 2 \to 3 \to 4 \to 6 = 9$

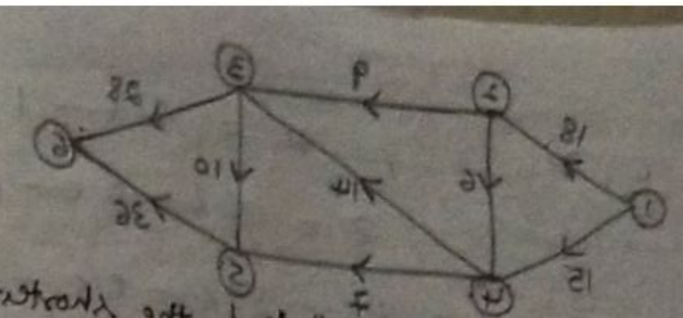
⑤ $1 \to 2 \to 3 \to 4 \to 6 \to 5 = 10$

⑥ $1 \to 2 \to 3 \to 4 \to 6 \to 5 \to 7 = 16$.

* Graph:-



Scanned By Camera Scanner



Using "Dijkstra's algorithm" find the shortest path from vertex '1' to other vertices (2, 3, 4, 5 and 6).

Sol^n:-

First Iteration:-

$P = \{1\}$, $t_j = d_{1j}$ $j = 2, 3, 4, 5, 6$

$t_2 = d_{12} = \infty$

$t_3 = d_{13} = \infty$

$t_4 = d_{14} = 15$

$t_5 = d_{15} = \infty$

$t_6 = d_{16} = \infty$

Step-01:- Among new t_j the minimum occurs for $j = 4$ the minimum being $t_4 = 15$.

Step-02:- $P = \{1, 4\}$ and $t_4 = 15$. Also $t_2 = \infty$, $t_3 = 18$, $t_5 = \infty$, $t_6 = \infty$

$k = 4$

new $t_j = \min\{t_j, t_k + d_{kj}\}$

new $t_j = \min\{t_j, t_4 + d_{4j}\}$ $j = 2, 3, 5, 6$.

new $t_2 = \min\{t_2, t_4 + d_{42}\} = \min\{\infty, 15 + \infty\} = \infty$

new $t_3 = \min\{t_3, t_4 + d_{43}\} = \min\{\infty, 15 + 10\} = 25$

new $t_5 = \min\{t_5, t_4 + d_{45}\} = \min\{\infty, 15 + 4\} = 19$

new $t_6 = \min\{t_6, t_4 + d_{46}\} = \min\{\infty, 15 + \infty\} = \infty$

Second Iteration:-

$P = \{1, 4\}$ and $t_4 = 15$. Also we have $t_2 = \infty$, $t_3 = 25$, $t_5 = 19$, $t_6 = \infty$

Step-01:- Among new t_j the minimum occurs for $j = 5$ the minimum being $t_5 = 19$

Step-02:- $P = \{1, 4, 5\}$ and $t_4 = 15$, $t_5 = 19$. Also we change

$k = 5$

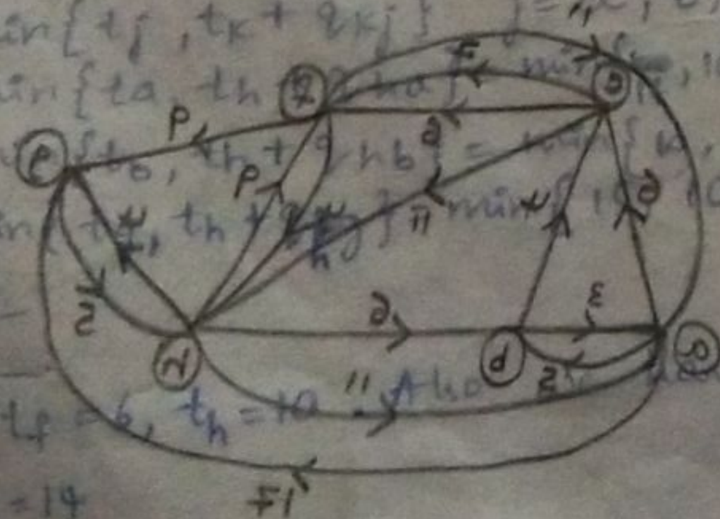
new $t_j = \min\{t_j, t_k + d_{kj}\}$ $j = 3, 2, 6$.

new $t_2 = \min\{t_2, t_5 + d_{52}\} = \min\{\infty, 19 + \infty\} = \infty$

new $t_3 = \min\{t_3, t_5 + d_{53}\} = \min\{25, 19 + 9\} = 25$

new $t_6 = \min\{t_6, t_5 + d_{56}\} = \infty$

Using "Dijkstra's algorithm" find the shortest path from vertex '1' to other vertices (a, b, f, d, e).



Third Iteration:-

$P = \{1, 4, 5\}$ and $t_4 = 15$, $t_5 = 18$. Also we have $t_2 = 25$, $t_3 = \infty$, $t_6 = \infty$.

Step-01:- Among new t_j the minimum occurs for $j = 2$ the minimum being $t_2 = 25$

Step-02:- $P = \{1, 4, 5, 2\}$ and $t_4 = 15$, $t_5 = 18$, $t_2 = 25$ we change

$k = 2$

new $t_j = \min\{t_j, t_k + d_{kj}\}$ $j = 3, 6$

new $t_3 = \min\{t_3, t_2 + d_{23}\} = \min\{25, 25 + \infty\} = 25$

new $t_6 = \min\{t_6, t_2 + d_{26}\} = \infty$

Fourth Iteration:-

$P = \{1, 4, 5, 2\}$ and $t_4 = 15$, $t_5 = 18$, $t_2 = 25$. Also we have $t_3 = 25$, $t_6 = 28$

Step-01:- Among new t_j the minimum occurs for $j = 3$ the minimum being $t_3 = 25$.

Step-02:- $P = \{1, 4, 5, 2, 3\}$ and $t_4 = 15$, $t_5 = 18$, $t_2 = 25$, $t_3 = 25$ we change ~~the value of t_6 as 28~~

$k = 3$

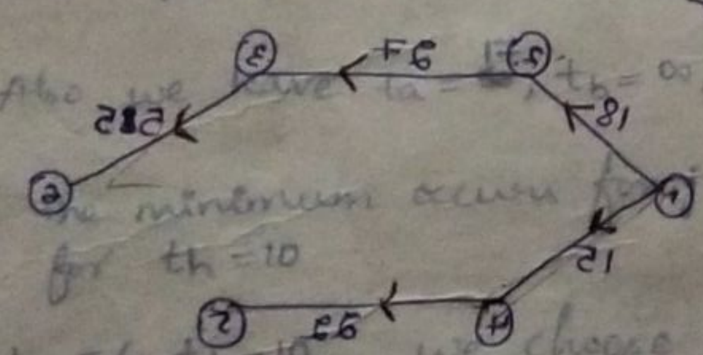
new $t_j = \min\{t_j, t_k + d_{kj}\}$ $j = 6$

new $t_6 = \min\{t_6, t_3 + d_{36}\} = \min\{28, 25 + \infty\} = 28$

Final Iteration:-

$P = \{1, 4, 5, 2, 3, 6\}$, $t_4 = 15$, $t_5 = 18$, $t_2 = 25$, $t_3 = 25$, $t_6 = 28$.

Graph:-



Scanned By Camera Scanner

Sol^n:-

c	a	b	f	g	h
1	2	3	4	5	6

First Iteration:-

$P = \{c\}$, $t_j = q_{cj}$ $\quad j = a, b, f, g, h$

$t_a = q_{ca} = \infty$

$t_b = q_{cb} = \infty$

$t_f = q_{cf} = 6$

$t_g = q_{cg} = \infty$

$t_h = q_{ch} = 11$

Step-01:- Among new t_j the minimum occurs for $j = f$, the minimum being $t_f = 6$

Step-02:- $P = \{c, f\}$ and $t_f = 6$. Also $t_a = \infty$, $t_b = \infty$, $t_g = \infty$, $t_h = 11$

$k = f$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $\quad j = a, b, g, h$

$t_a = \min\{t_a, t_f + q_{fa}\} = \min\{\infty, 6 + 11\} = 17$

$t_b = \min\{t_b, t_f + q_{fb}\} = \min\{\infty, 6 + \infty\} = \infty$

~~$t_d = \min\{t_d, t_f + q_{fd}\} = \min\{$~~

~~$t_e = \min\{t_e, t_f + q_{fe}\} = \min\{$~~

$t_g = \min\{t_g, t_f + q_{fg}\} = \min\{\infty, 6 + 9\} = 15$

$t_h = \min\{t_h, t_f + q_{fh}\} = \min\{11, 6 + 4\} = 10$

Second Iteration:-

$P = \{c, f\}$ and $t_f = 6$. Also we have $t_a = 17$, $t_b = \infty$, $t_g = 15$, $t_h = 10$

Step-01:- Among new t_j the minimum occurs for $j = h$, the minimum being for $t_h = 10$

Step-02:- $P = \{c, f, h\}$ and $t_f = 6$, $t_h = 10$. we choose

$k = h$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $\quad j = a, b, g$

new $t_a = \min\{t_a, t_h + q_{ha}\} = \min\{17, 10 + 11\} = 17$

new $t_b = \min\{t_b, t_h + q_{hb}\} = \min\{\infty, 10 + 6\} = 16$

new $t_g = \min\{t_g, t_h + q_{hg}\} = \min\{15, 10 + 4\} = 14$

Third Iteration:-

$P = \{c, f, h\}$ and $t_f = 6$, $t_h = 10$. Also we have $t_a = 17$, $t_b = 16$, $t_g = 14$

Step-01:- Among new t_j the minimum occurs for $j = g$, the minimum being for $t_g = 14$

Step-02:-

$P = \{c, f, h, g\}$ and $t_f = 6$, $t_h = 10$, $t_g = 14$. we choose

$k = g$

new $t_j = \min\{t_j, t_k + q_{kj}\} \Rightarrow j = a, b$

new $t_a = \min\{t_a, t_g + q_{ga}\} = \min\{17, 14 + \infty\} = 17$

new $t_b = \min\{t_b, t_g + q_{gb}\} = \min\{16, 14 + \infty\} = 16$

Fourth Iteration:-

$P = \{c, f, h, g\}$ and $t_f = 6$, $t_h = 10$, $t_g = 14$. Also we have $t_a = 17$, $t_b = 16$.

Step-01:- Among new t_j the minimum occurs for $j = b$, the minimum being for $t_b = 16$

Step-02:-

$P = \{c, f, h, g, b\}$ and $t_f = 6$, $t_h = 10$, $t_g = 14$, $t_b = 16$ we choose

$k = b$

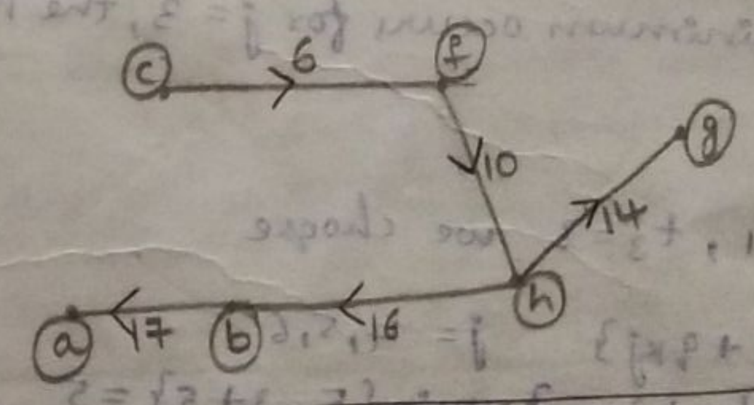
new $t_j = \min\{t_j, t_k + q_{kj}\}$ $\quad j = a$

new $t_a = \min\{t_a, t_b + q_{ba}\} = \min\{17, 16 + 3\} = 17$

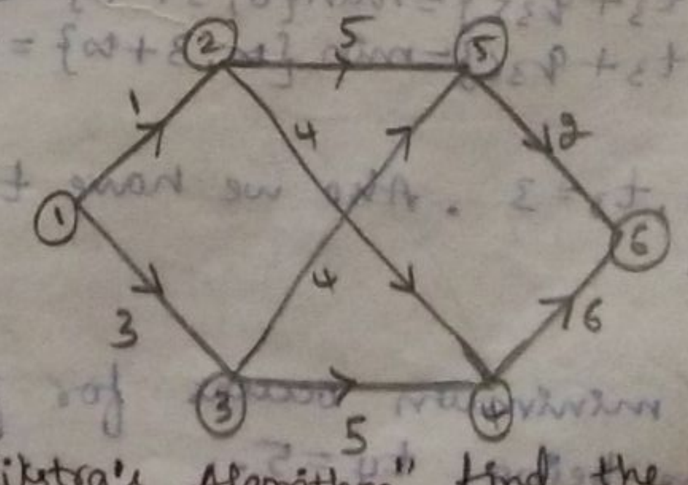
Final Iteration:-

$P = \{c, f, h, g, b, a\}$, $t_f = 6$, $t_h = 10$, $t_g = 14$, $t_b = 16$, $t_a = 17$

Graph:-



(H)



Using "Dijkstra's Algorithm" find the shortest path from vertex

"1" to other vertices (2, 3, 4, 5 and 6).

Soln:- First Iteration:-

$P = \{1\}$ $t_j = q_{1j}$ $j = 2, 3, 4, 5, 6$

$t_2 = q_{12} = 1$

$t_3 = q_{13} = 3$

$t_4 = q_{14} = \infty$

$t_5 = q_{15} = \infty$

$t_6 = q_{16} = \infty$

Step - 01:-

Among new t_j the minimum occurs for $j = 2$ the minimum being $t_2 = 1$.

Step - 02:-

$P = \{1, 2\}$ and $t_2 = 1$. Also $t_3 = 3$, $t_4 = \infty$, $t_5 = \infty$, $t_6 = \infty$

$k = 2$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 3, 4, 5, 6$

new $t_3 = \min\{t_3, t_2 + q_{23}\} = \min\{3, 1+\infty\} = 3$

new $t_4 = \min\{t_4, t_2 + q_{24}\} = \min\{\infty, 1+4\} = 5$

new $t_5 = \min\{t_5, t_2 + q_{25}\} = \min\{\infty, 1+5\} = 6$

new $t_6 = \min\{t_6, t_2 + q_{26}\} = \min\{\infty, 1+\infty\} = \infty$

Second Iteration:-

$P = \{1, 2\}$ and $t_2 = 1$. Also we have $t_3 = 3$, $t_4 = 5$, $t_5 = 6$, $t_6 = \infty$

Step - 01:-

Among new t_j the minimum occurs for $j = 3$, the minimum being $t_3 = 3$.

Step - 02:-

$P = \{1, 2, 3\}$ and $t_2 = 1$, $t_3 = 3$ we choose

$k = 3$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 4, 5, 6$

new $t_4 = \min\{t_4, t_3 + q_{34}\} = \min\{5, 3+5\} = 5$

new $t_5 = \min\{t_5, t_3 + q_{35}\} = \min\{6, 3+4\} = 6$

new $t_6 = \min\{t_6, t_3 + q_{36}\} = \min\{\infty, 3+\infty\} = \infty$

Third Iteration:-

$P = \{1, 2, 3\}$ and $t_2 = 1$, $t_3 = 3$. Also we have $t_4 = 5$, $t_5 = 6$, $t_6 = \infty$.

Step - 01:-

Among new t_j the minimum occurs for $j = 4$ the minimum being $t_4 = 5$.

Step - 02:-

$P = \{1, 2, 3, 4\}$ and $t_2 = 1$, $t_3 = 3$, $t_4 = 5$ we choose

$k = 4$

new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 5, 6$

new $t_5 = \min\{t_5, t_4 + q_{45}\} = \min\{6, 5+\infty\} = 6$

new $t_6 = \min\{t_6, t_4 + q_{46}\} = \min\{\infty, 5+6\} = 11$

Fourth Iteration:-

$P = \{1, 2, 3, 4\}$ and $t_2 = 1$, $t_3 = 3$, $t_4 = 5$ and also $t_5 = 6$, $t_6 = 11$

Step - 01:-

Among new t_j the minimum occurs for $j = 5$, the minimum being $t_5 = 6$.

Step - 02:-

$P = \{1, 2, 3, 4, 5\}$ and $t_2 = 1$, $t_3 = 3$, $t_4 = 5$, $t_5 = 6$ we choose

$k = 5$

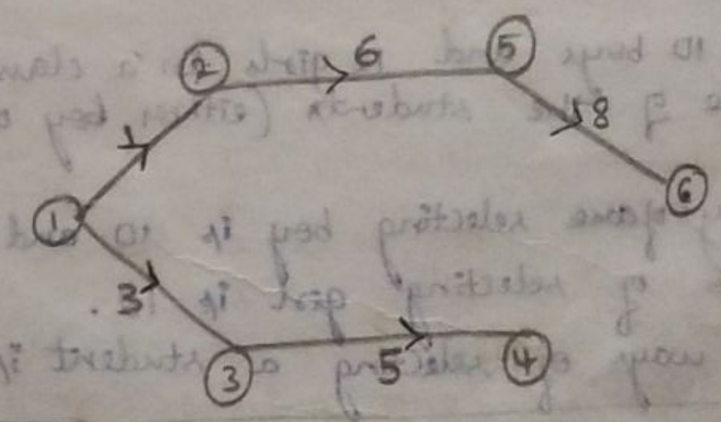
new $t_j = \min\{t_j, t_k + q_{kj}\}$ $j = 6$

new $t_6 = \min\{t_6, t_5 + q_{56}\} = \min\{11, 6+2\} = 8$

Final Iteration:-

$P = \{1, 2, 3, 4, 5, 6\}$ and $t_2 = 1$, $t_3 = 3$, $t_4 = 5$, $t_5 = 6$, $t_6 = 8$.

Graph:-



Scanned By Camera Scanner