MODULE-5
TREES

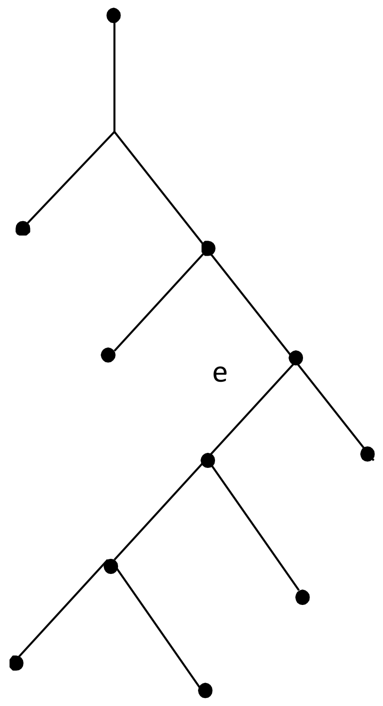
DISCRETE MATHEMATICS & GRAPH THEORY

21CIDS31

TREES

- Connected graph without circuits is called a **tree.**
- Graph is called a **forest** when it does not have circuits.
- A vertex of degree 1 is called a terminal vertex or a **leaf**, the other vertices are called internal **nodes**.

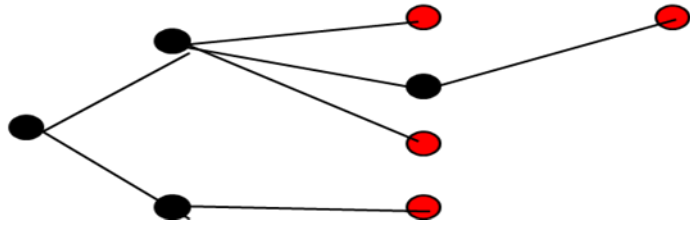
Examples: Decision tree, Syntactic derivation tree.



Note:

- Any tree with more than one vertex has at least one vertex of degree 1.
- Any tree with n vertices has n – 1 edges.

  That is every tree T = (V, E), | V | = | E | + 1 (or | E | = | V | - 1).
- If a connected graph with n vertices has n – 1 edges, then it is a tree.



Examples:

If a tree T has 3 vertices of degree 2, 2 vertices of degree 3 and 2 vertices of degree 4. Find the number of pendant vertices in T.

Solution:

Let N be the number of pendant vertices in T.

The total number of vertices in the tree

T = N + 3 + 2 + 2 = N + 7

Therefore the number of edges in the tree

T = N + 7 – 1 = N + 6

By handshaking property sum of

all the degrees of the vertices:

(By handshaking lemma $\sum_{i=1}^{p} deg(v_i) = 2q$)

(N ×1) + (3 × 2) + (2 × 3) + (2 × 4)

= 2(N + 6)

N + 6 + 6 + 8 = 2N + 12

2N – N = 20 – 12

N = 8.

Example:

If a tree T has four vertices of degree 2, one vertex of degree 3, two vertices of degree 4, one vertex of degree 5. Find the number of leaves in T.

Solution:

Let N be the number of pendant vertices in T.

The total number of vertices in the tree

T = N + 4 + 1 + 2 + 1 = N + 8

Therefore the number of edges in the tree T = N + 8 – 1 = N + 7

By handshaking property sum of all the degrees of the vertices: (By handshaking lemma $\sum_{i=1}^{p} deg(v_i) = 2q$)

(N × 1) + (4 × 2) + (1 × 3) + (2 × 4) + (1 × 5) = 2(N + 7)

N + 8 + 3 + 8 + 5 = 2N + 14

2N – N = 24 – 14

N = 10.

Example:

Suppose that a tree T has two vertices of degree 2, four vertices of degree 3 and three vertices of degree 4. Find the number of pendant vertices in T.

Solution:

Let N be the number of pendant vertices in T.

The total number of vertices in the tree T = N + 2 + 4 + 3 = N + 9

Therefore the number of edges in the tree T = N + 9 – 1 = N + 8

By handshaking property sum of all the degrees of the vertices:

(By handshaking lemma $\sum_{i=1}^{p} deg(v_i) = 2q$)

(N × 1) + (2 × 2) + (4 × 3) + (3 × 4) = 2(N + 8)

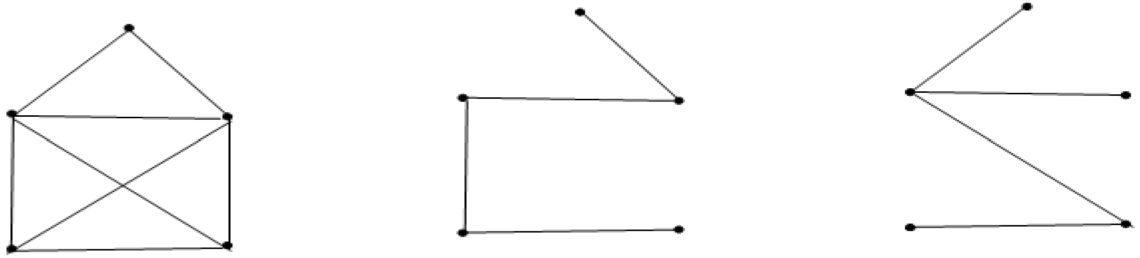
N + 4 + 12 + 12 = 2N + 16

2N – N = 28 – 16

N = 12.

Spanning Trees:

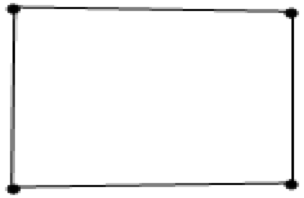
A subgraph T of a graph G is called a spanning tree when T is a tree and contains all vertices of G. Every connected graph has a spanning tree. Any two spanning trees have the same number of edges. A weighted graph is a graph in which each edge has an associated real number weight. A minimal spanning tree (MST) is a spanning tree with the least total weight of its edges.



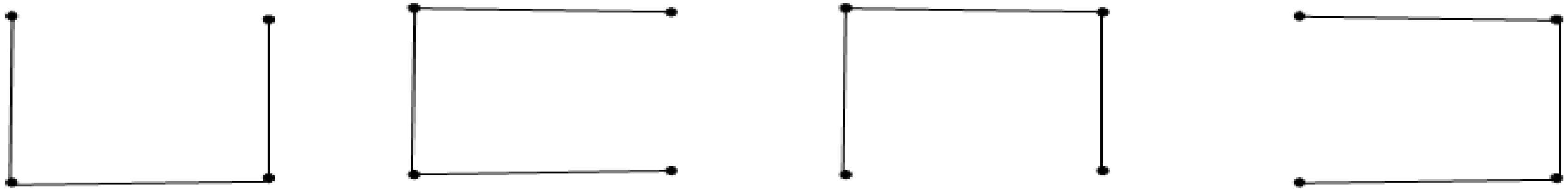
G T1 T2

Example:

Find the spanning trees of the graph



Solution:

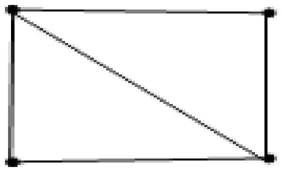


Note:-

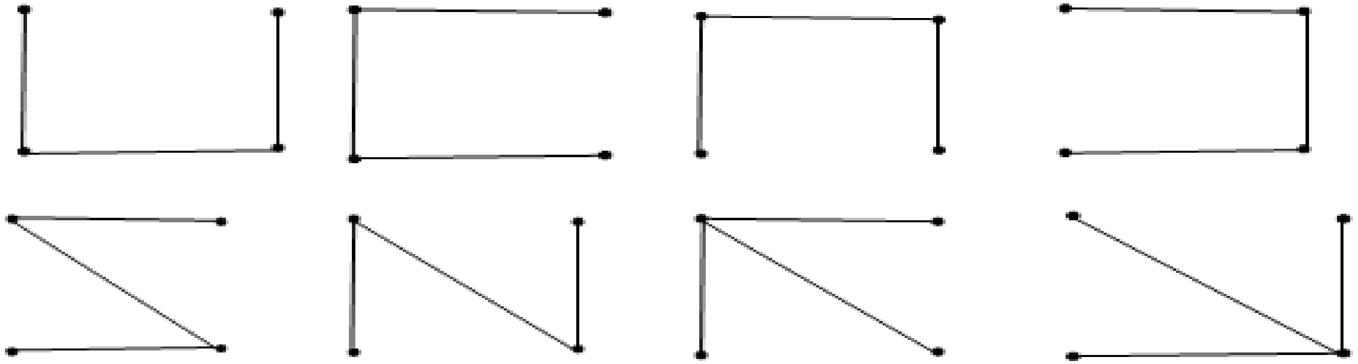
Prove that G = (V, E) is an undirected graph then G is connected if and only if G has a spanning tree.

Example:

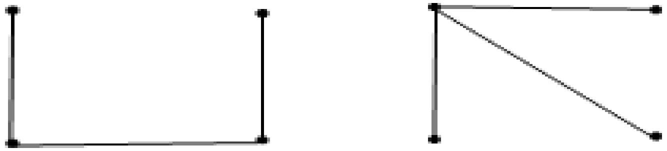
Find the spanning trees of the graph. Find all the non-isomorphic spanning trees of the graph G



Solution:

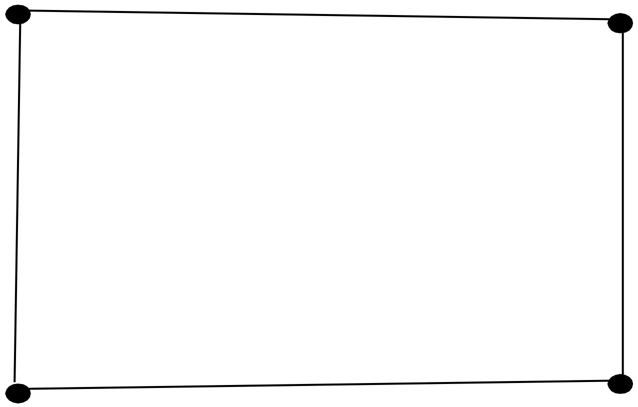


Non isomorphic trees

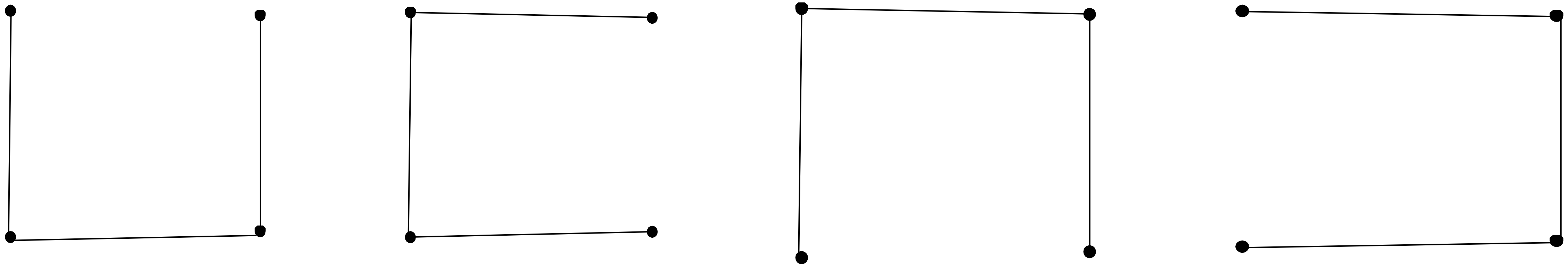


Example:

Find the spanning trees of the graph



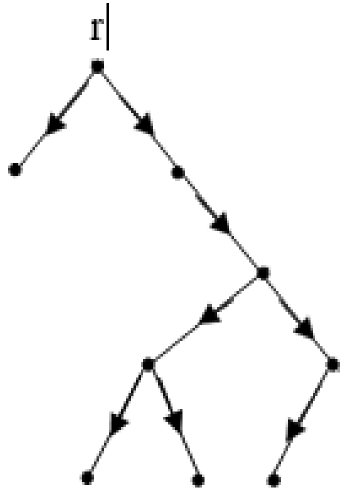
Solution:

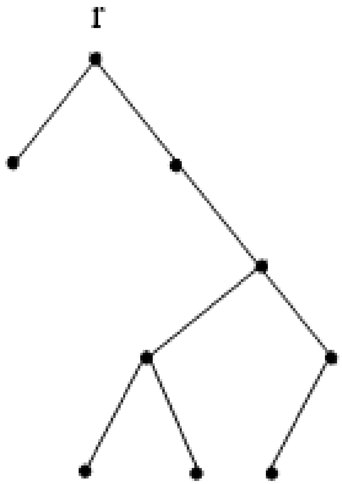


Rooted Tree:

A directed graph of tree T is said to be the rooted tree if T contains a unique vertex whose in degree is zero. In a rooted tree a vertex whose in degree is zero is called as root of the tree. The in degree of all other vertices of T is equal to one.

In a rooted tree we denote a root by ‘r’ and draw the tree down wards from upper level to lower level so that arrows can be dropped, then the root t will be at the uppermost level and all other vertices will be at the lower level.





Internal vertices:

Vertices in a tree except pendent vertices are called as internal vertices.

m –ary tree:

A rooted tree T is said to be m-ary tree if the out degree of the every internal vertex of T is less than or equal to m. A rooted tree T is said to be complete m-ary tree if the out degree of the every internal vertex of T is exactly m.

Binary Tree:

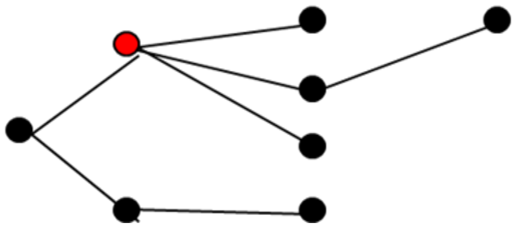
A rooted tree T is said to be binary tree if the out degree of internal vertices of T is less than or equal to 2.

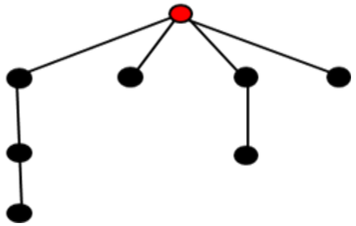
Complete Binary Tree:

A rooted tree T is said to be complete binary tree if the out degree of internal vertices of T is equal to 2.

Rooted Trees:

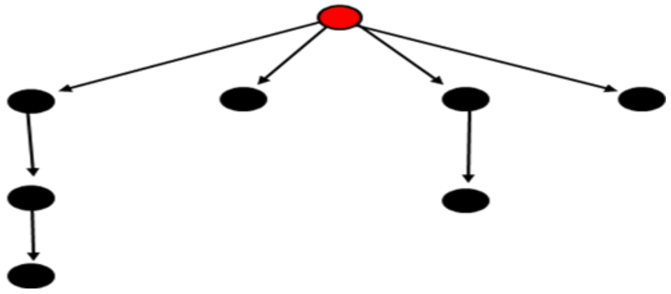
- Rooted tree is a tree in which one vertex is distinguished and called a root. Level of a vertex is the number of edges between the vertex and the root. The height of a rooted tree is the maximum level of any vertex. Children, siblings and parent vertices in a rooted tree. Ancestor, descendant relationship between vertices





Rooted Directed Trees:

It is sometimes useful to turn a rooted tree into a rooted directed tree T′ by directing every edge away from the root.

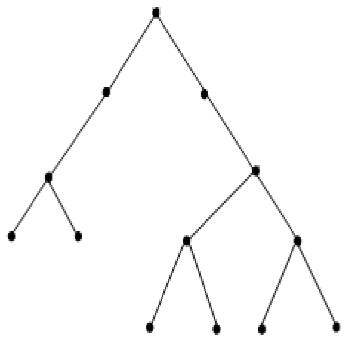


Binary Trees:

- Binary tree is a rooted tree where each internal vertex has at most two children: left and right. Left and right subtrees.
- Full binary tree: Representation of algebraic expressions
- If T is a full binary tree with k internal vertices then T has a total of 2k + 1 vertices and k + 1 of them are leaves. Any binary tree with t leaves and height h satisfies the following inequality: $t \leq 2h$

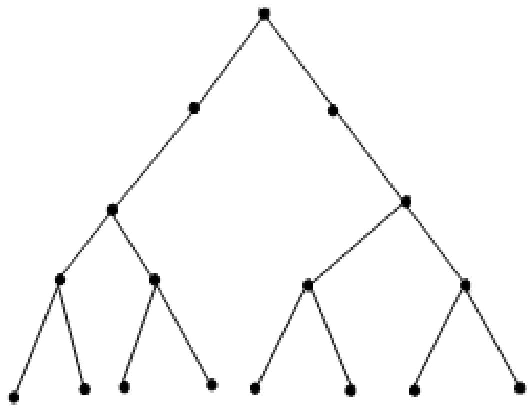
Balanced Tree:

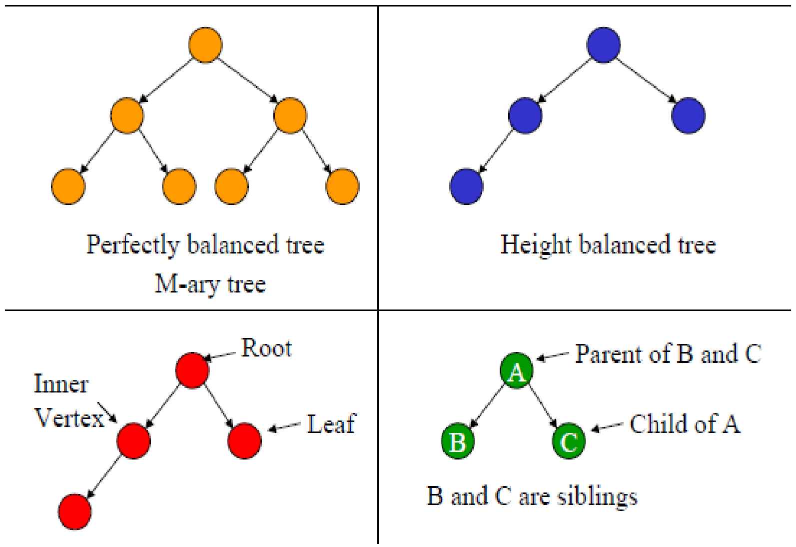
A rooted tree T of height h is said to be balanced tree if the level number of the leaf is h or h-1.



Full Balanced Tree:

A rooted tree T is called full balanced tree if all the leaves in the tree are at the same level





Note:

Let T be a complete m-ary tree of order n with p leaves and q internal vertices then we have

a) $n = mq + 1 = \frac{mp-1}{m-1}$

b) $p = (m-1)q + 1 = \frac{(m-1)n+1}{m}$

c) $q = \frac{n-1}{m} = \frac{p-1}{m-1}$

Examples:

Find the number of vertices and the number of leaves in a complete binary tree having 10 internal vertices.

Solution:

Given q = 10, m = 2, n = ?, p = ? $n = mq + 1$

$n = mq + 1$

$n = (2.10) + 1$

= 21

$p = (m - 1)q + 1$

= (2 – 1)10 + 1

= 11.

Example:

A class room has 25 micro computers that has to be connected to a wall socket that has four outlets. The connections are made by using extension cords that have four outlets each. What is the least number of cords needed to get computer to use.

Solution:

Here p = 25, m = 4, q = ?

$$q = \frac{p - 1}{m - 1}$$

$$q = \frac{25 - 1}{4 - 1}$$

$$q = \frac{24}{3}$$

q = 8

Therefore the number of cords required are = 8 – 1 = 7.

Example:

A computer laboratory of a school has 10 computers that are to be connected to the wall socket that has two outlets. The connections are made by using extension cords that have two outlets each. Find the least number of cords needed to get these computer setup for use.

Solution:

Here p = 10, m = 2, q = ?

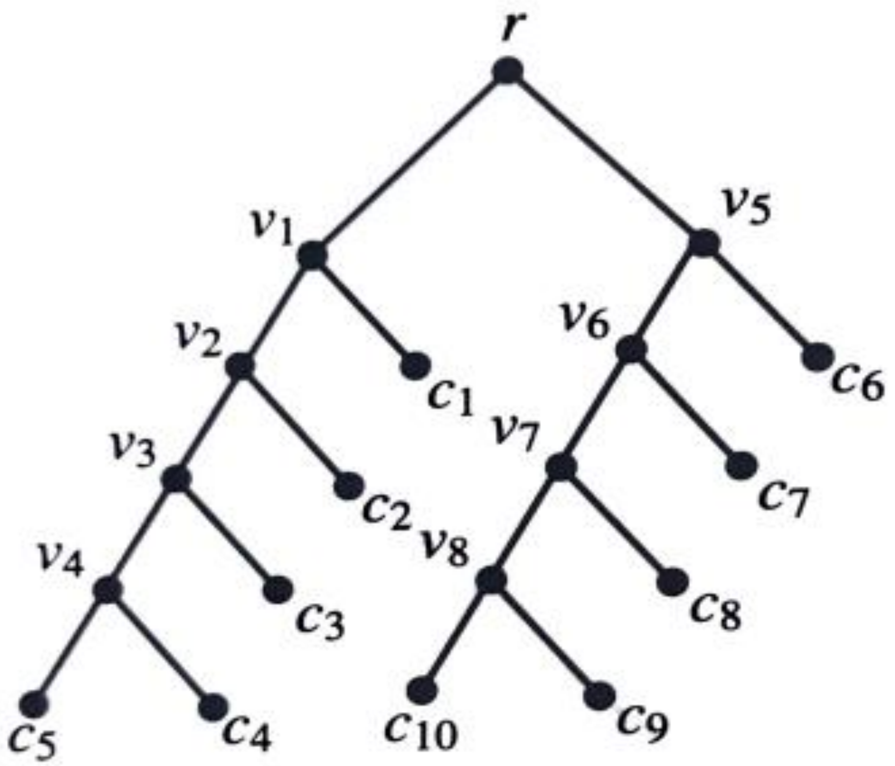
$$q = \frac{p-1}{m-1}$$

$$q = \frac{10-1}{2-1}$$

$$q = \frac{9}{1}$$

$$q = 9$$

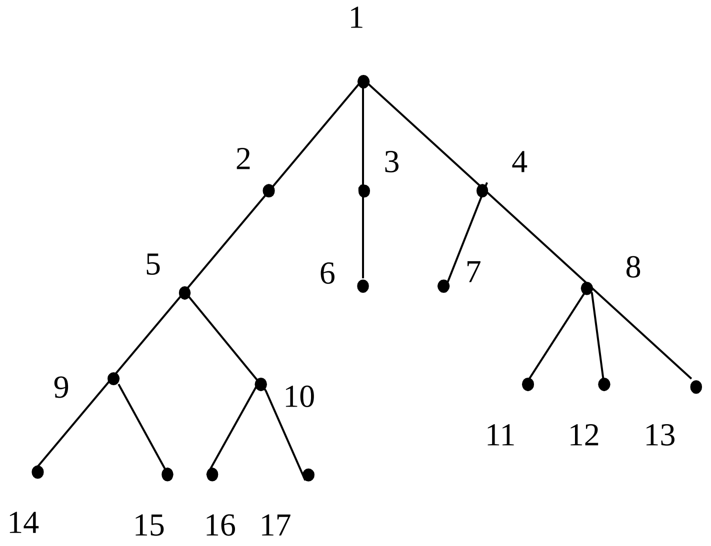
Therefore the number of cords required are = 9 – 1 = 8.



Preorder and Postorder Traversals:

Example:

List the vertices in the given tree, when they are visited in the reorder and postorder traversels.



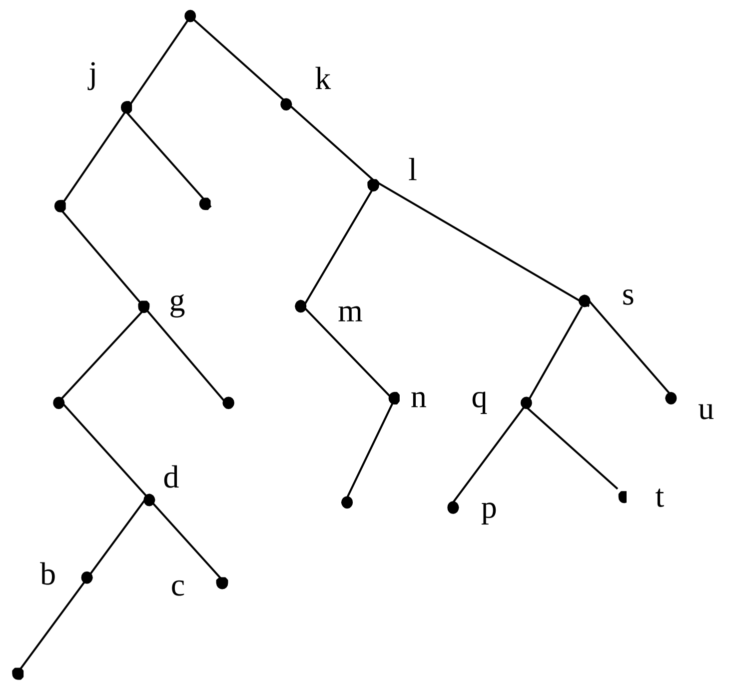
Solution:

Preorder:1, 2, 5, 9, 14, 15, 10, 16, 17, 3, 6, 4, 7, 8, 11, 12, 13.

Postorder:14, 15, 9, 16, 17, 10, 5, 2, 6, 3, 7, 11, 12, 13, 8, 4, 1.

Example:

List the vertices in the given tree, when they are visited in the reorder and postorder traversels.



Preorder: r, j, h, g, e, d, b, a, c, f,i, k, l, m, n, o, s, q, p, t, u.

Postorder: a, b, c, d, e, f, g, h, i, j, o, n, m, p, t, q, u, s, l, k, r.

Sorting:

Explain the steps in the merge sort algorithm.

Suppose we wish to sort (rearrange/reorganise) a given list of n integers in nondecreasing order.

The most common (and the easiest) way of carrying out this sorting consists of two parts.

In the first part, we recursively split the given list and all subsequent lists in half (or as close as possible to half) until each sublist contains a single element.

In the second part, we merge the sublists in increasing order until the original n integers have been sorted.

The splitting and merging process is done by the use of balanced complete tree. This method of sorting a list is known as Merge sort.

Example:

Using the merge sorting method sort the list {12, 11, 2, 3, 1, 5, 6, 7, 10, 9, 4, 15}

Solution:

{12, 11, 2, 3, 1, 5, 6, 7, 10, 9, 4, 15}

{12, 11, 2, 3, 1, 5} {6, 7, 10, 9, 4, 15}

{12, 11, 2} {3, 1, 5} {6, 7, 10} {9, 4, 15}

{12, 11} {2} {3, 1} {5} {6, 7} {10} {9, 4} {15}

{12} {11} {3} {1} {6} {7} {9} {4}

{11, 12} {2} {1, 3} {5} {6, 7} {10} {4, 9} {15}

{2, 11, 12} {1, 3, 5} {6, 7, 10} {4, 9, 15}

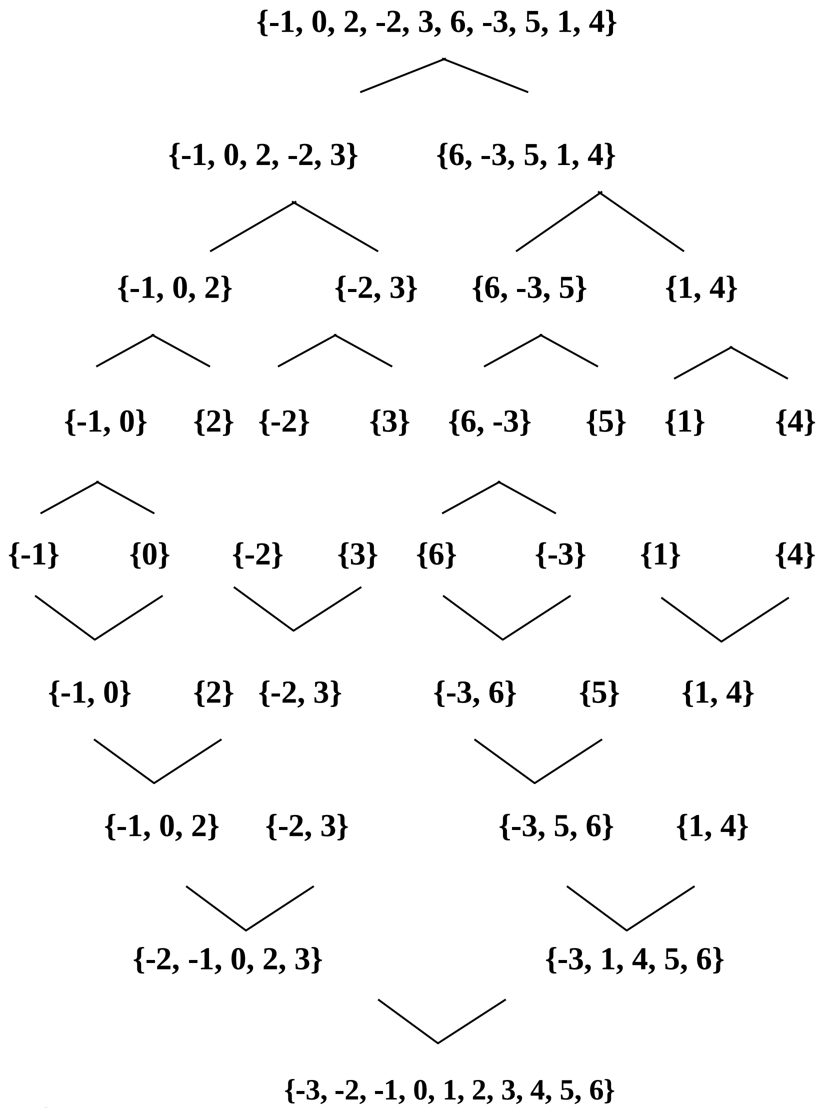
{1, 2, 3, 5, 11, 12} {4, 6, 7, 9, 10, 15}

{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 15}

Example:

Using the merge sorting method sort the following list -1, 0, 2, -2, 3, 6, -3, 5, 1, 4

Solution:



Prefix codes:

A set P of binary sequence is called prefix code if no sequence in P is the prefix of any other sequence in P.

For example, the sets

$P_1 = \{000, 001, 01, 10, 11\}$

$P_2 = \{10, 0, 1101, 111, 1100\}$

are prefix codes,

where as the sets

$A_1 = \{01, 0, 101, 10, 1\}$

$A_2 = \{1, 00, 01, 000, 0001\}$

are not prefix codes.

Example:

Consider the prefix codes codes

a : 111, b : 0 c : 1100, d : 1101, e : 10

using these codes decode the following sequences

(i) 1001111101 (ii) 1101111110010 (iii) 10111100110001101

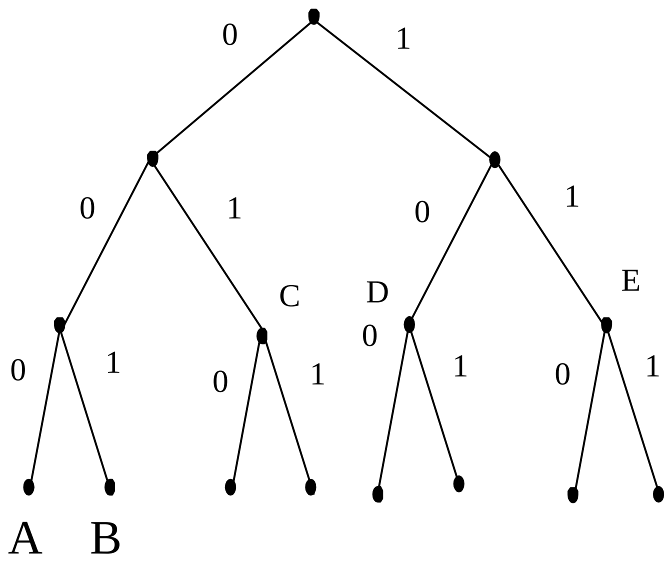
Solution:

(i) e b a d (ii) d a c e (iii) e a e b c b d

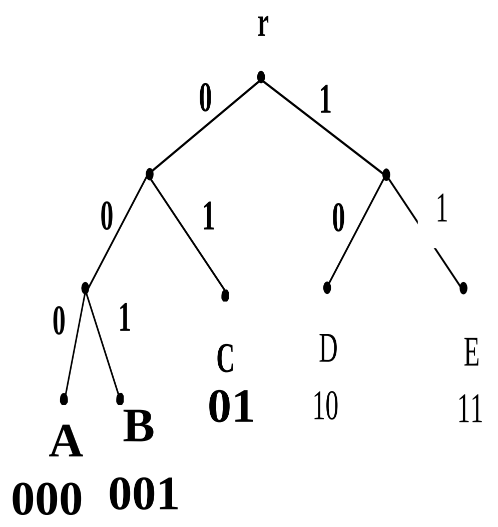
Prefix codes can be represented by binary trees as illustrated below:

Consider the prefix code P = {000, 001, 01, 10, 11}

In this code, the longest sequence has length 3. Keeping this in mind, let us construct a full binary tree of height 3, and assign the symbol 0 to every edge that is directed towards the child in the left from its parent vertex and 1 to every edge that is directed towards the child in the right, as shown in figure

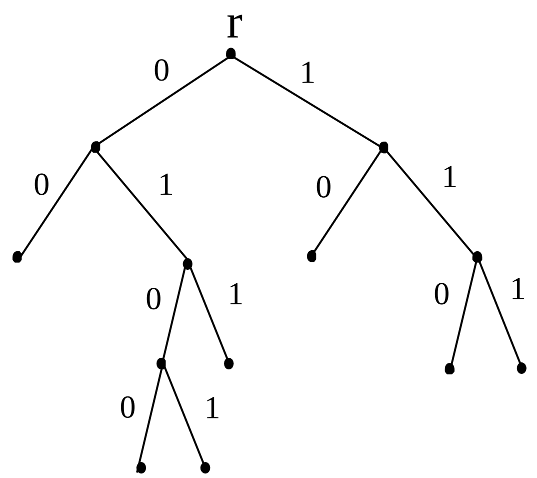


The five sequences present in P_1 can now be identified with five vertices of the above tree. We note that the vertex marked A can be reached from the root r through the three edges 0, 0,
0. Accordingly, the vertex A can be assigned the sequence 000. Similarly the vertex marked B can be assigned the sequence 001. The vertex marked C can be the sequence 01, the vertex marked D can be assigned the sequence 10, and the vertex marked E can be assigned the sequence 11. Thus, all the five sequences present in P_1 can be assigned to the five vertices, marked A, B, C, D, E, of the tree being considered.
The subtree extracted from the full binary tree from the above figure that contains the root r and the vertices A, B, C, D, E is shown in the below figure. This sub tree represent the prefix code given by the set P_1.

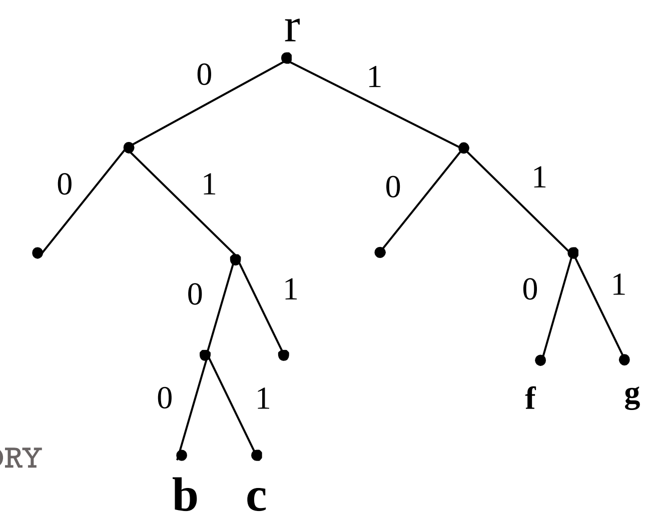


Example:

Find the prefix code represented by the following tree



Solution:



The leaves of the given tree are represented by the symbols a, b, c, d, e, f, g. These leaves are identified by the sequences as indicated in the following Table

Leaf	a	b	c	d	e	f	g
Sequence	00	0100	0101	011	10	110	111

Weighted Tree:

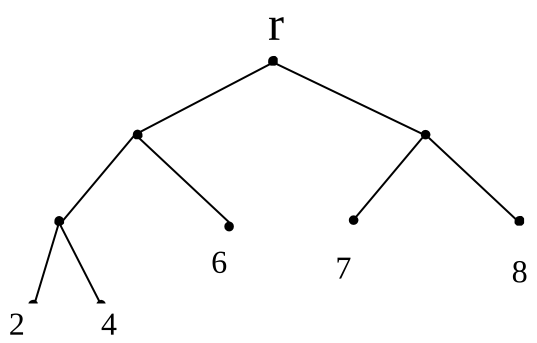
Consider a set of n positive integers $w_1, w_2, w_3, \ldots..w_n$ where $w_1 \leq w_2 \leq w_3 \leq w_4 \leq .. \leq w_n$. Suppose we assign these integers to the n leaves of a complete binary tree T = (V, E) in any one-to-one manner. The resulting tree is called a complete weighted, binary tree with $w_1, w_2, w_3, \ldots..w_n$ as weights. If $l(w_i)$ is the level number of the leaf of T to which the weight w_i is assigned, then W(T) defined by

$$W(T) = \sum_{i=1}^{n} w_i l(wi)$$

is called the weight of the tree T.

Example:

Find weight of the following tree T



Solution:

Given set of integers is {2, 4, 6, 7, 8}

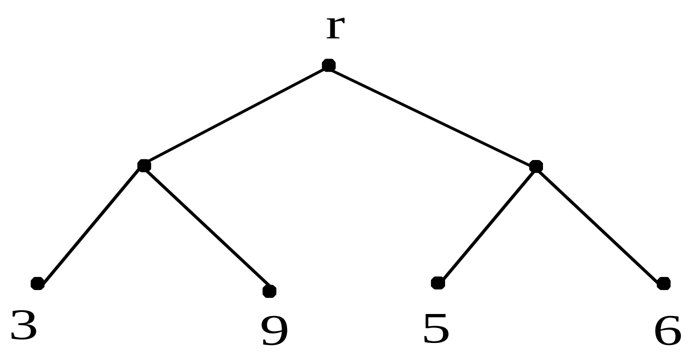
W(T) = (2.3) + (4.3) + (6.2) + (7.2) + (8.2)

= 6 + 12 + 12 + 14 + 16

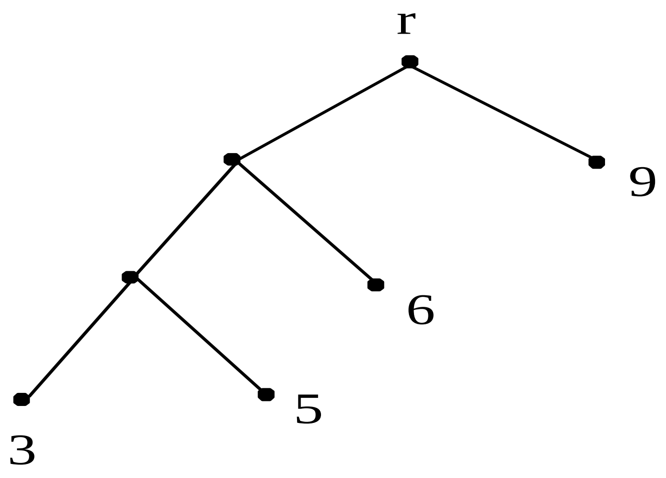
= 60

Example:

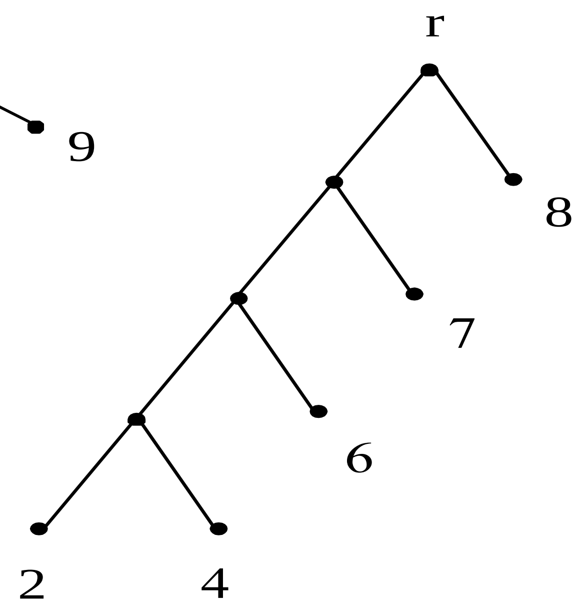
Find weight of the following trees



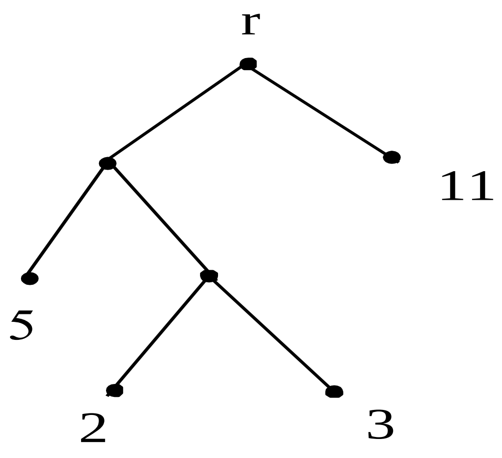
T_1



T_2



T_3



T_4

Optimal Tree: (Huffman Tree)

Optimal tree is a complete binary tree which carries the minimum weight for the given set of weights.

Example:

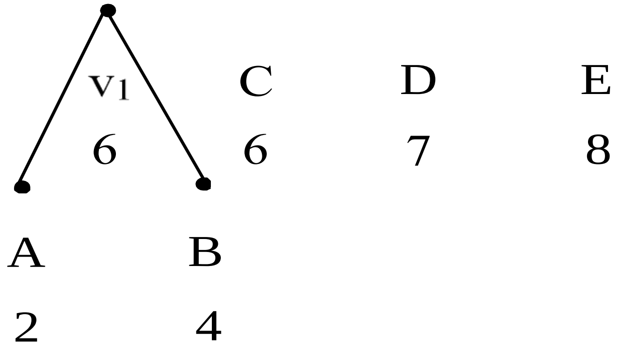
Find the optimal tree for the given set of weights

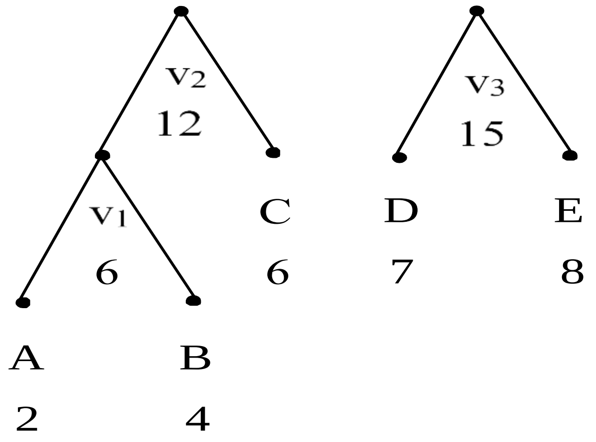
{2, 6, 7, 4, 8}

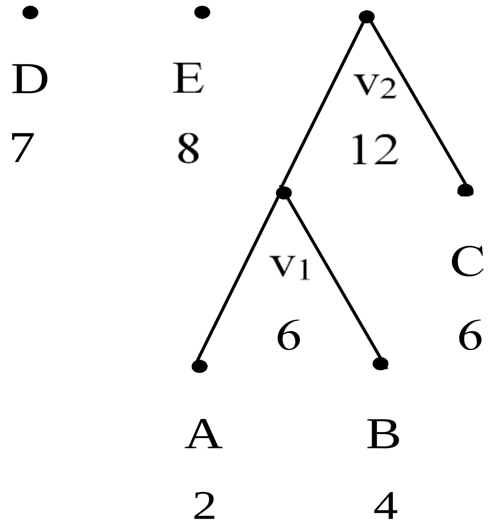
Arrange the given numbers in ascending order

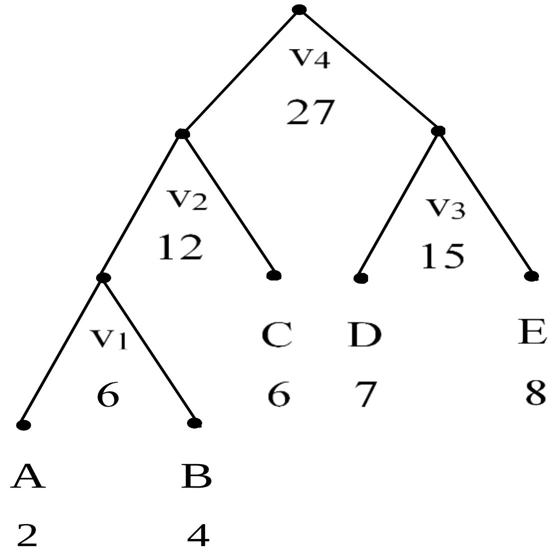
A	B	C	D	E
2	4	6	7	8

The vertices A and B carry the smallest weight, 2 and 4 add these weights to get the weight 6 assign it to the new vertex v_1. Draw a tree having v_1 as the root and A and B as its children. Rearrange the vertices present at this stage in the no decreasing order of their weights









$$W(T) = (2.3) + (4.3) + (6.2) + (7.2) + (8.2)$$

$$= 6 + 12 + 12 + 14 + 16 = 60$$

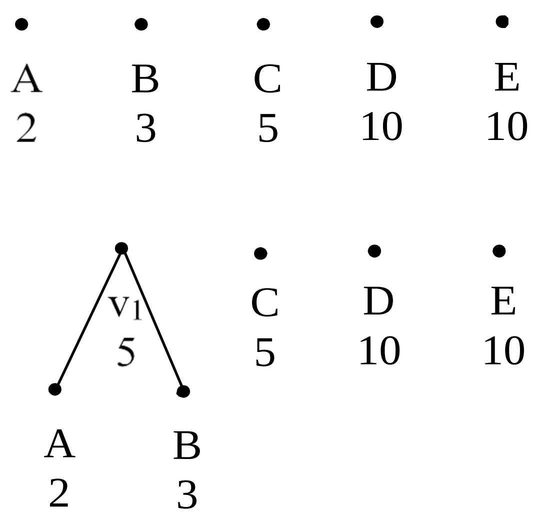
Example:

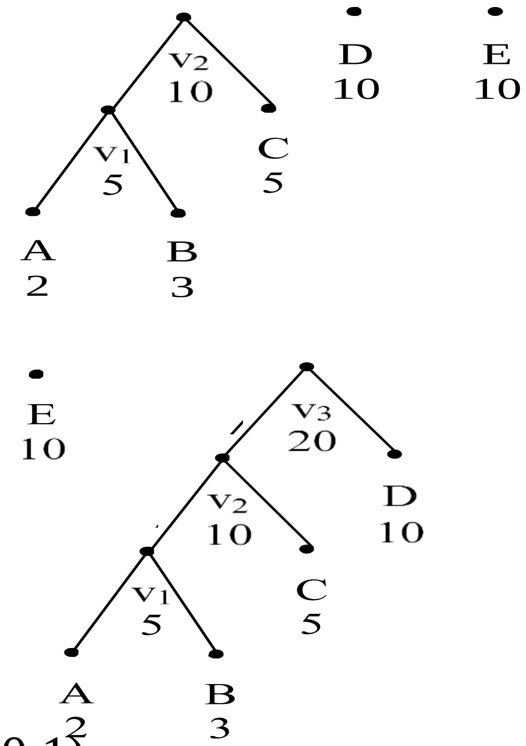
Construct an optimal tree for the given set of weights {4, 15, 25, 5, 8, 16}

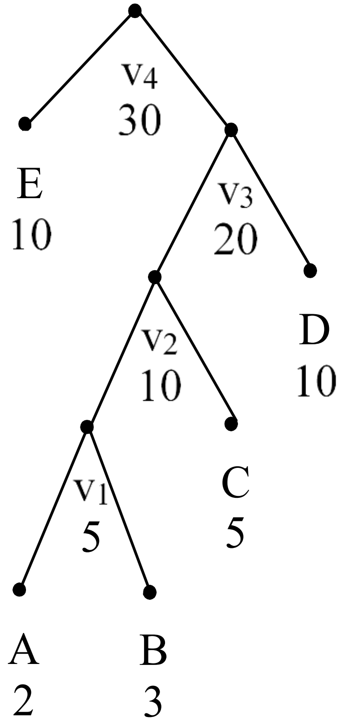
Example:

Using the weights 2, 3, 5, 10, 10, show that the height of the Huffman tree for a given set of weights is not unique.

Solution:

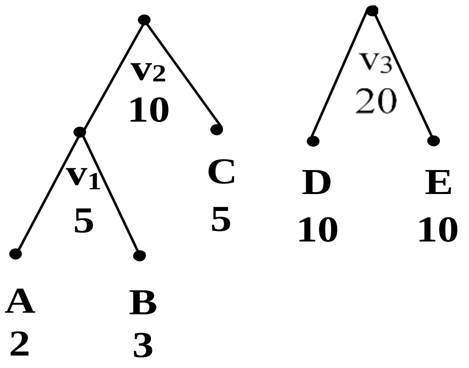


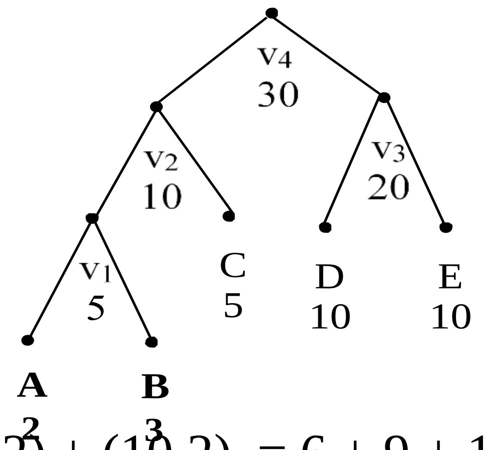




$W(T_1) = (2 .4) + (3.4) + (5.3) + (10.2) + (10.1)$

$= 8 + 12 + 15 + 20 + 10 = 65$



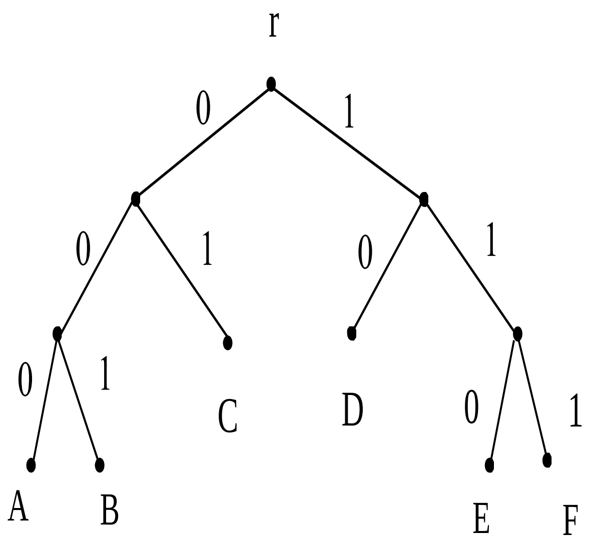


$W(T_2) = (2.3) + (3.3) + (5.2) + (10.2) + (10.2) = 6 + 9 + 10 + 20 + 20 = 65$

T_1 and T_2 are the two different optimal trees of the same set of weights. We find that both the trees are having the same weight but the different height. Therefore the height of the Huffman tree for a given set of weights is not unique.

Optimal Prefix Code:

The prefix codes which are obtained from the optimal tree are called optimal prefix codes. Since the optimal tree is not unique the optimal prefix code is also not unique.

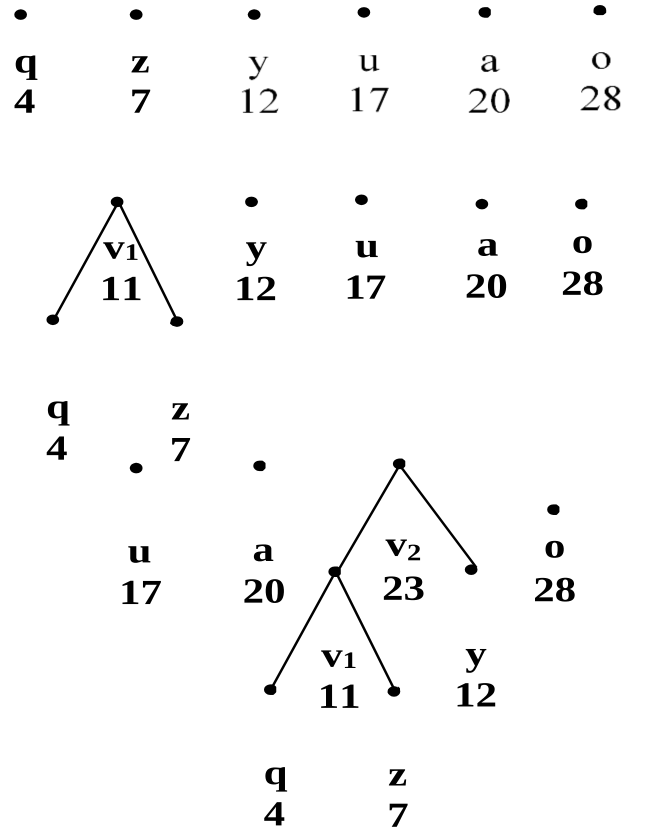


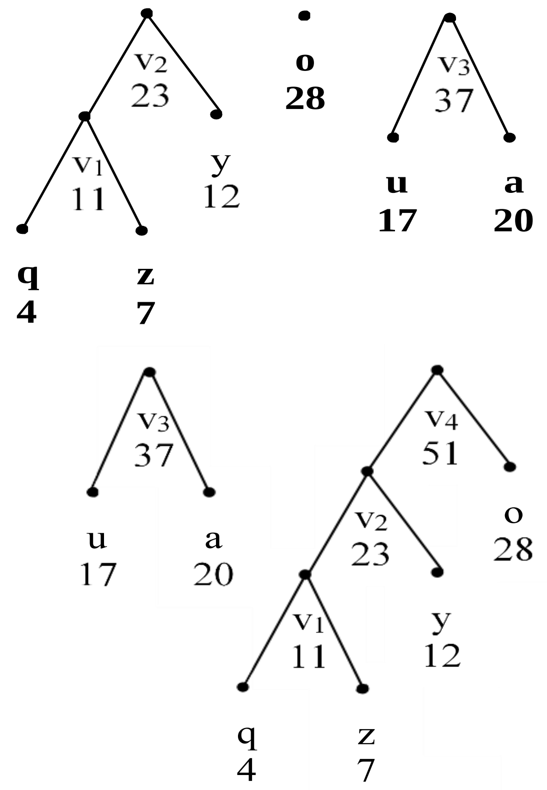
Vertices	A	B	C	D	E	F
Prefix code	**000**	**001**	**01**	**10**	**110**	**111**

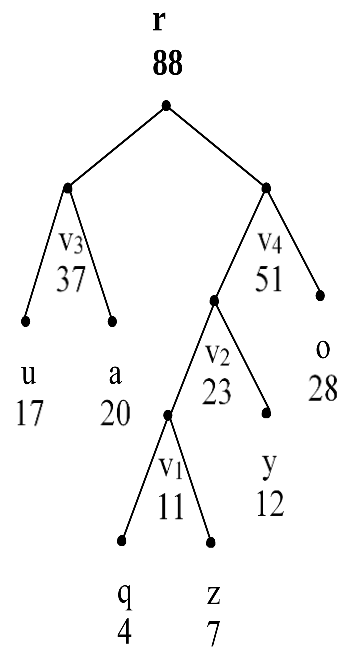
Example:

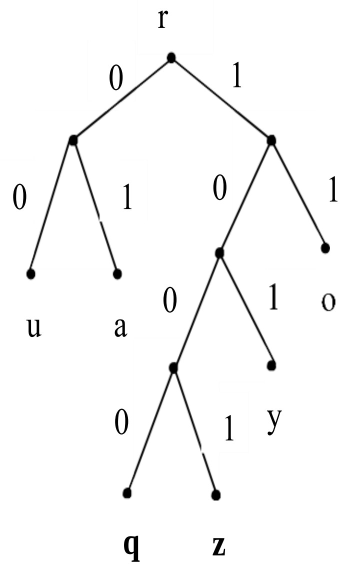
Construct the optimal prefix code for the symbols a, o, q, u, y, z that occurs with frequencies 20, 28, 4, 17, 12, 7 respectively.

Solution:









The Prefix codes are

Vertices	a	o	q	u	y	z
Prefix code	01	11	1000	00	101	1001

Example:

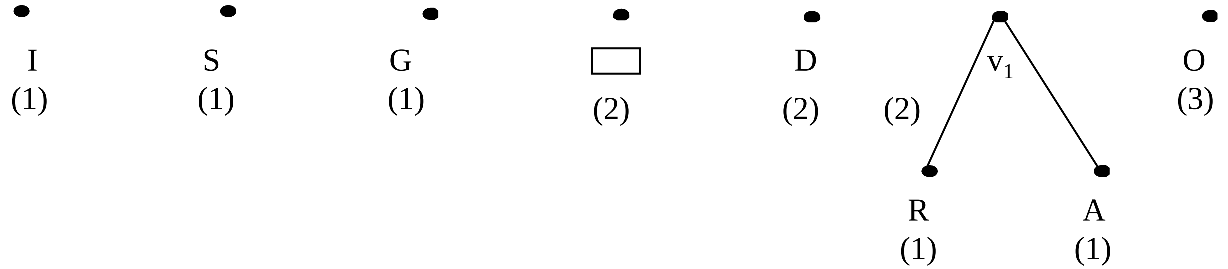
Construct the optimal prefix code for the message “ROAD IS GOOD”. Indicate the code.

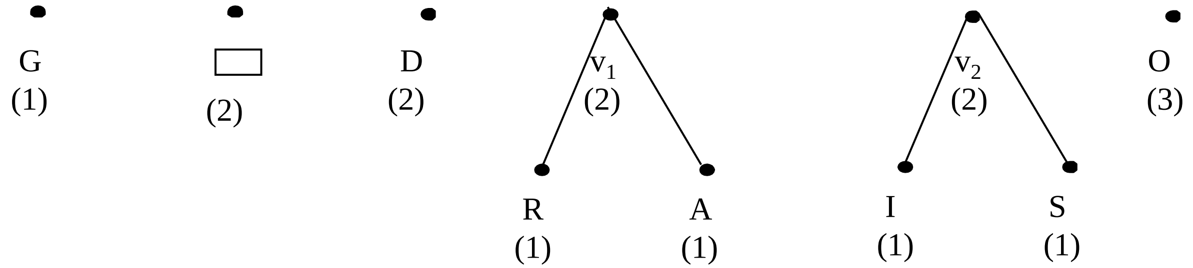
Solution:

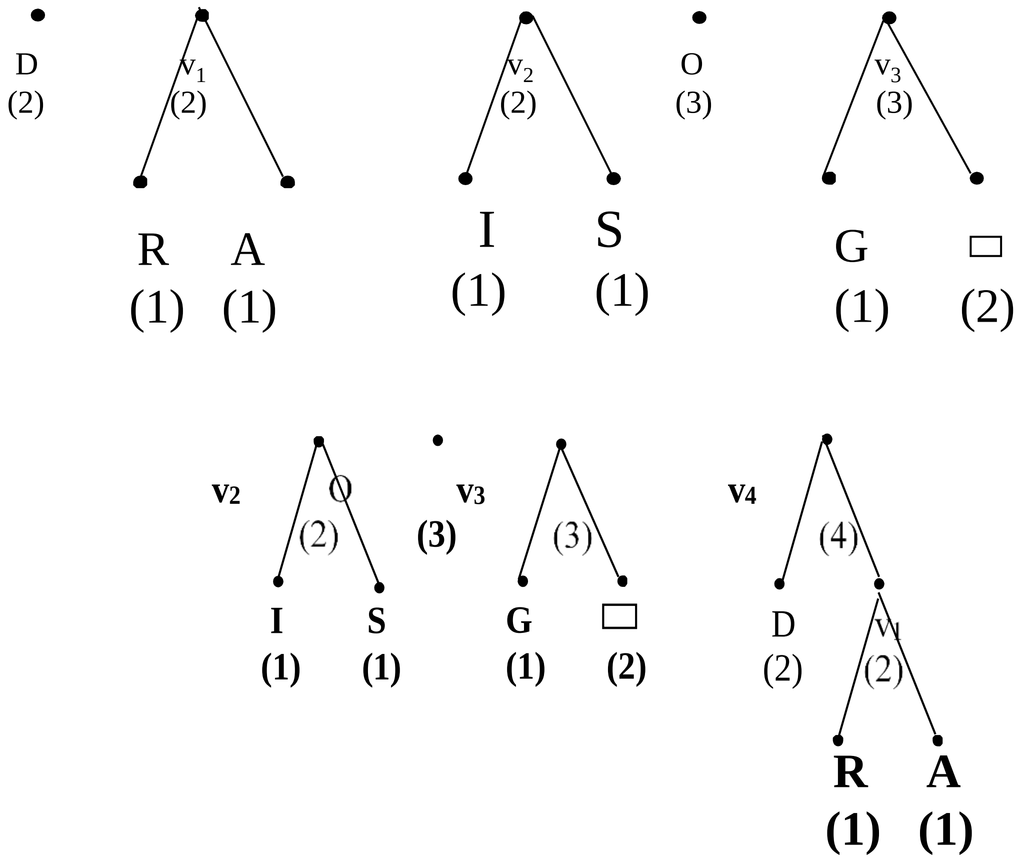
The given message consists of letters R, O, A, D, I, S, G with frequencies 1, 3, 1, 2, 1, 1, 1 respectively. Further, there is a blank space (□) occurring twice.

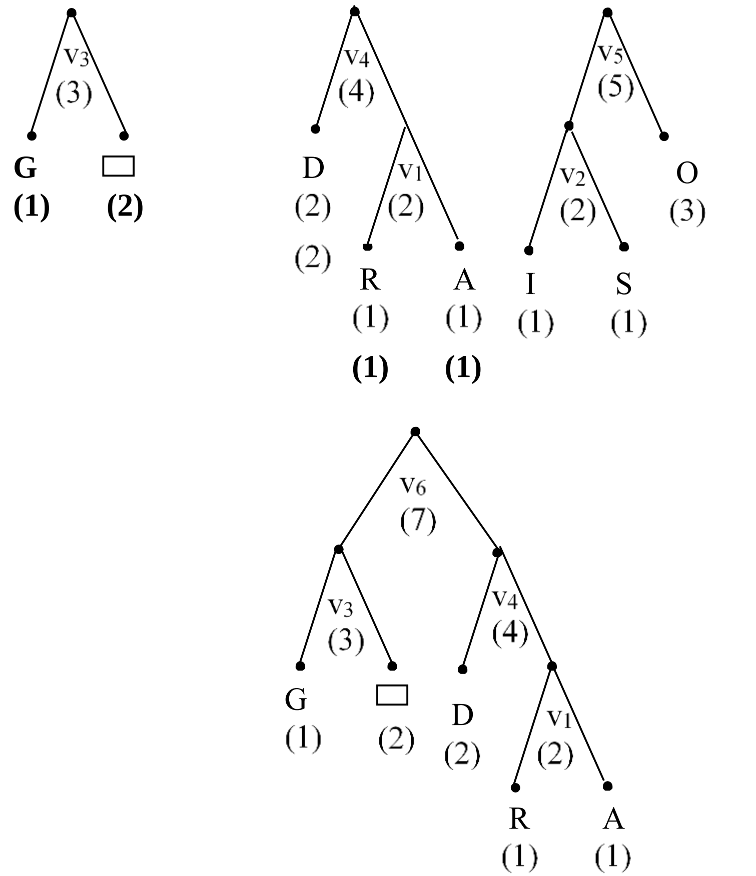
Arrange the symbols and □ in increasing order of their weights(frequencies), then construct the optimal tree using Huffman’s procedure.

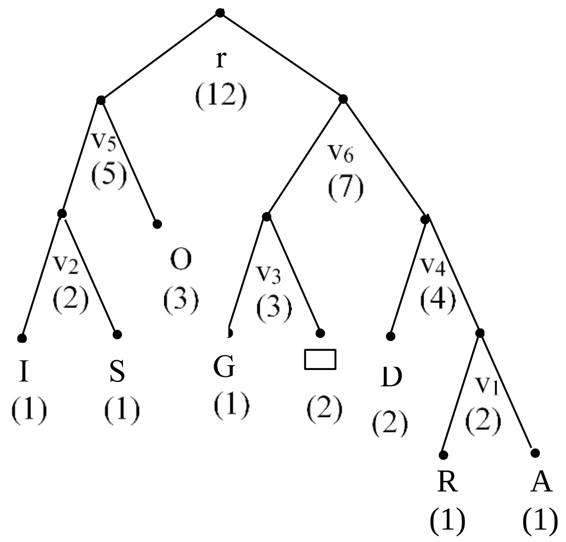
R	A	I	S	G	□	D	O
(1)	(1)	(1)	(1)	(1)	(2)	(2)	(3)

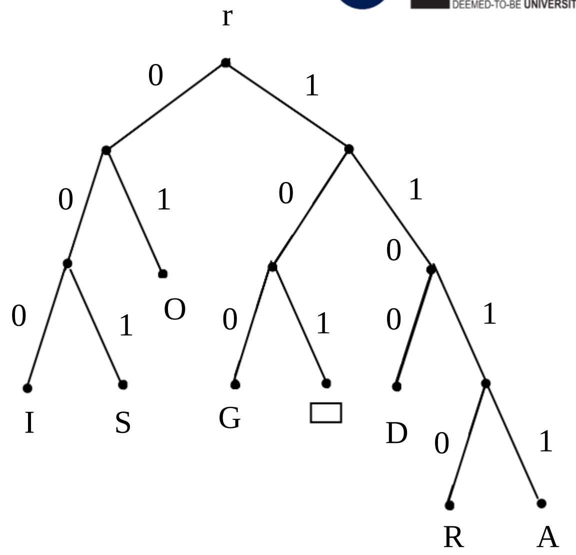





D
(2)
v1
(2)
R A
(1) (1)
v2
(2)
I S
(1) (1)
O
(3)
v3
(3)
G
(1) (2)
v2
O
(2)
I S
(1) (1)
(3)
v3
(3)
G
(1) (2)
v4
(4)
D
(2)
v1
(2)
R A
(1) (1)







Symbol	R	A	I	S	G	□	D	O
PrifixCode	1110	111	000	001	100	101	110	01

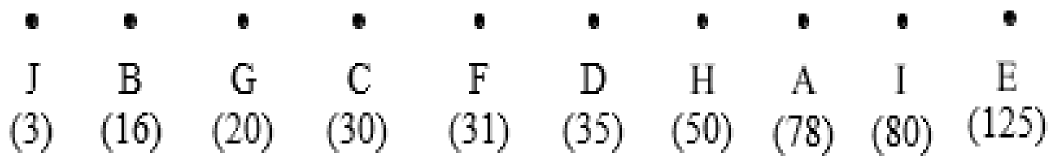
The code for the given message ROAD IS GOOD is 1110011111101010000011011000101110

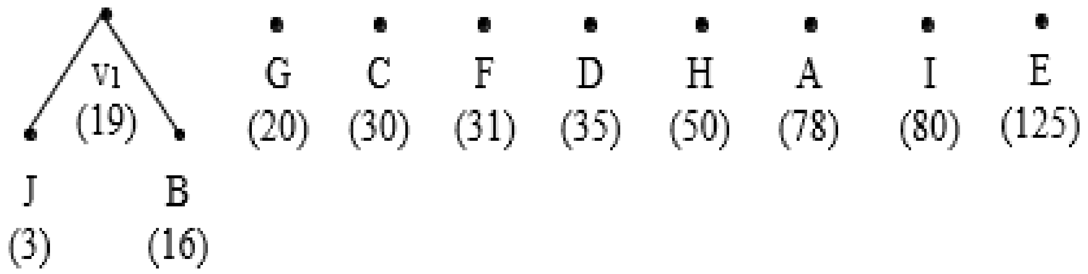
Example:

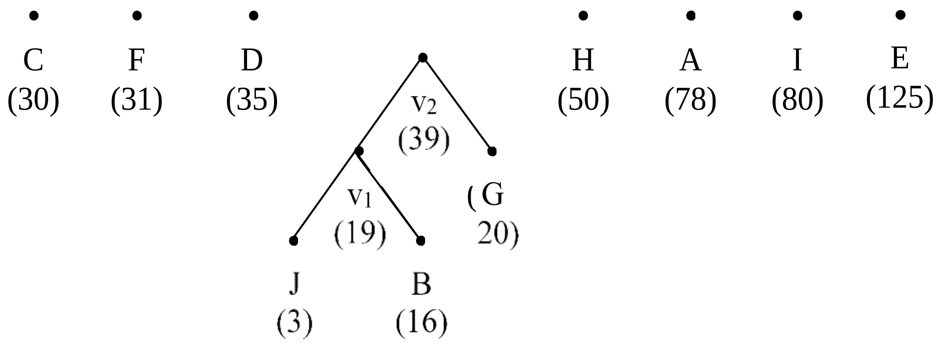
Construct the optimal prefix code for the symbols A, B, C, D, E, F, G, H, I, J that occurs with frequencies 78, 16, 30, 35, 125, 31, 20, 50, 80, 3 respectively.

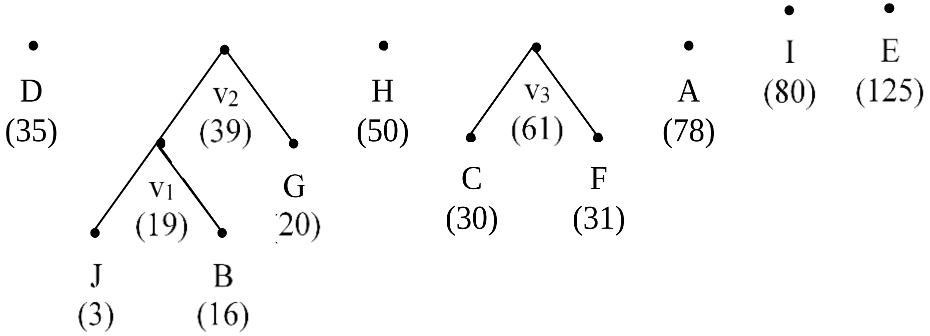
Solution:

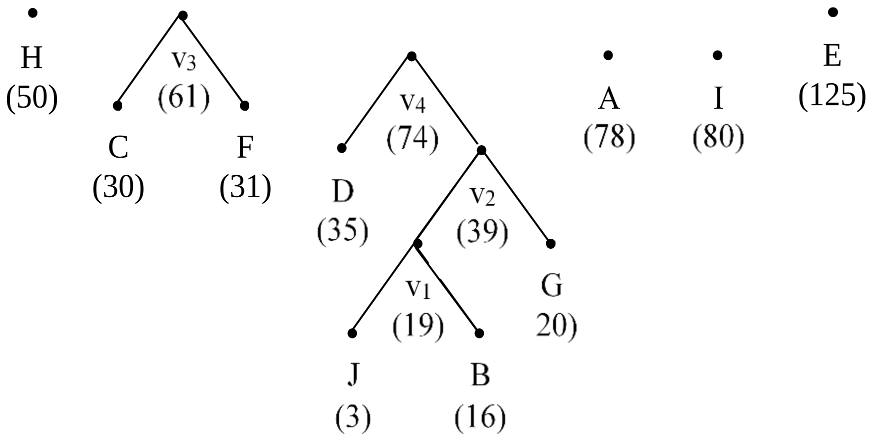
Arrange the symbols in increasing order of their weights then construct the optimal tree using Huffman's procedure.

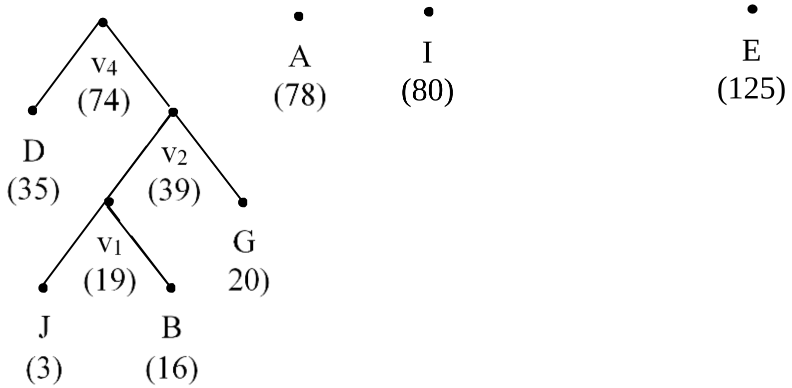


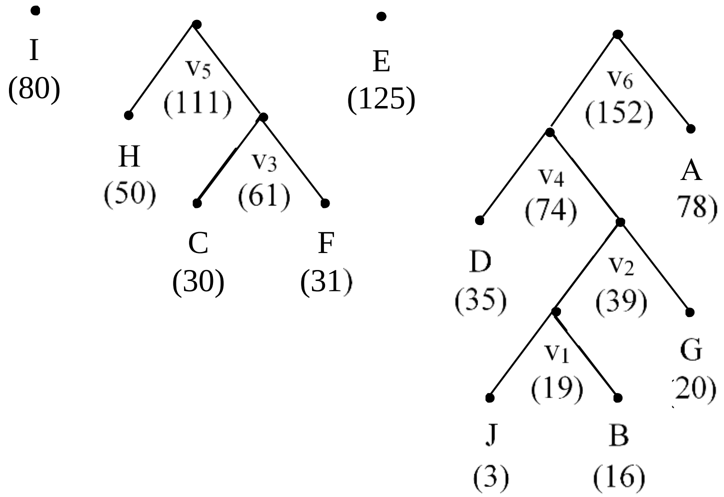


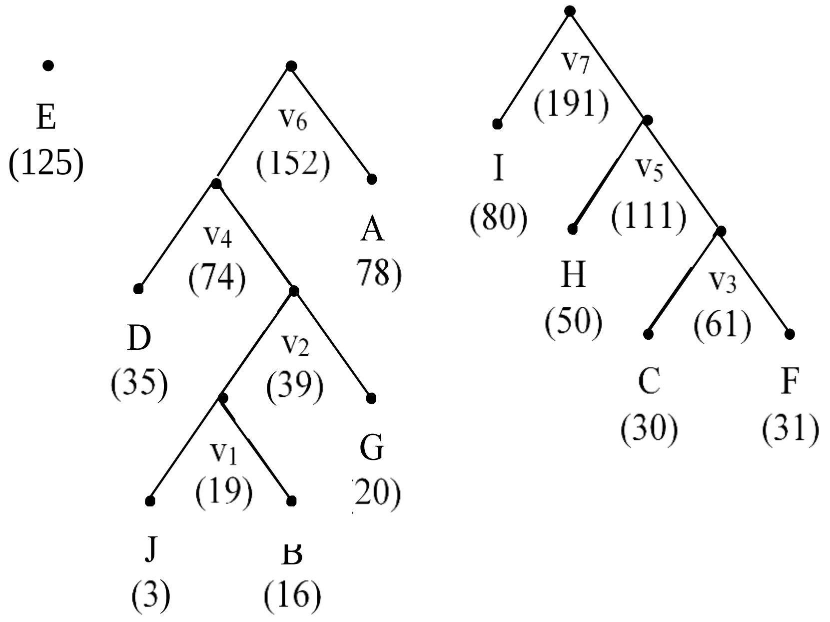

C (30)
F (31)
D (35)
v2 (39)
v1 (19)
J (3)
B (16)
G 20)
H (50)
A (78)
I (80)
E (125)

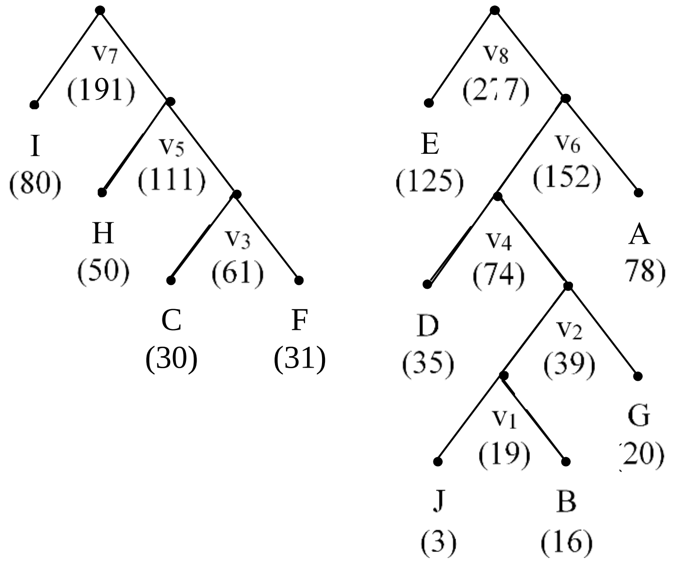

D (35)
v2 (39)
v1 (19)
J (3)
B (16)
G 20)
H (50)
v3 (61)
C (30)
F (31)
A (78)
I (80)
E (125)

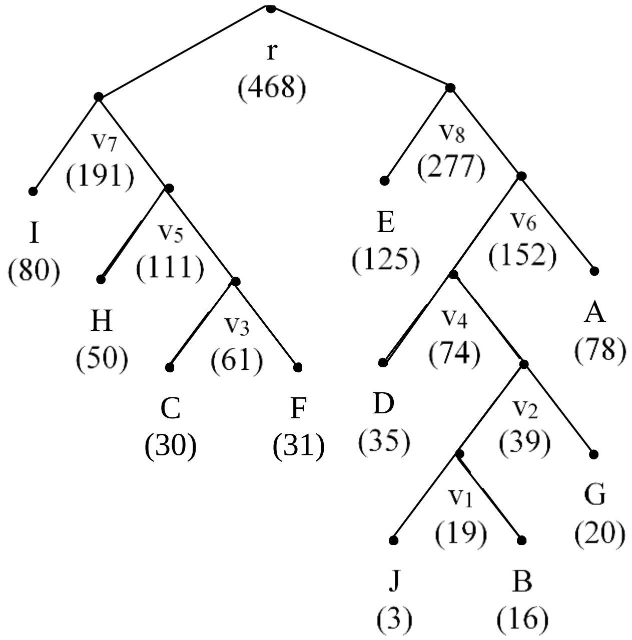

H (50)
v3 (61)
C (30)
F (31)
v4 (74)
D (35)
v2 (39)
v1 (19)
J (3)
B (16)
G 20)
A (78)
I (80)
E (125)

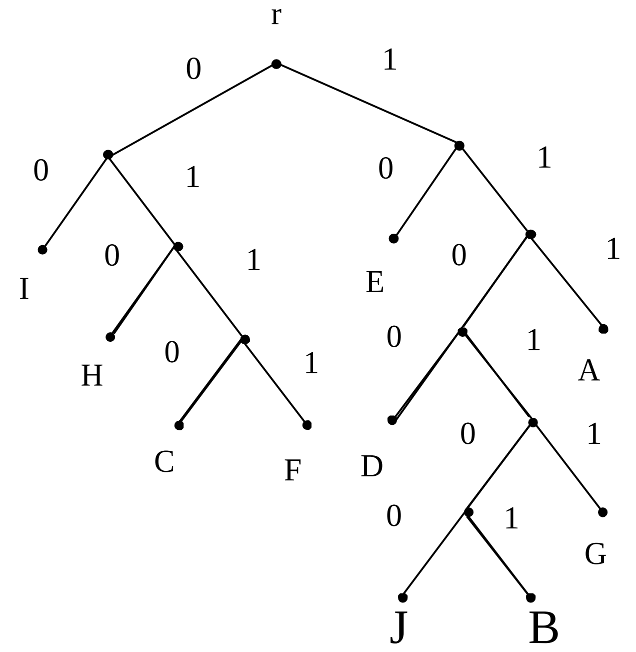

v4 (74)
D (35)
v2 (39)
v1 (19)
J (3)
B (16)
G 20)
A (78)
I (80)
E (125)


I (80)
v5 (111)
H (50)
v3 (61)
C (30)
F (31)
E (125)
v6 (152)
v4 (74)
A 78)
D (35)
v2 (39)
v1 (19)
G 20)
J (3)
B (16)


E
(125)
v6
(152)
A
78)
v4
(74)
D
(35)
v2
(39)
G
20)
v1
(19)
J
(3)
B
(16)
v7
(191)
I
(80)
v5
(111)
H
(50)
v3
(61)
C
(30)
F
(31)


v7
(191)
I
(80)
v5
(111)
H
(50)
v3
(61)
C
(30)
F
(31)
v8
(277)
E
(125)
v6
(152)
A
78)
v4
(74)
D
(35)
v2
(39)
G
20)
v1
(19)
J
(3)
B
(16)


r
(468)
v7
(191)
I
(80)
v5
(111)
H
(50)
v3
(61)
C
(30)
F
(31)
v8
(277)
E
(125)
v6
(152)
A
(78)
v4
(74)
D
(35)
v2
(39)
G
(20)
v1
(19)
J
(3)
B
(16)


r
0
1
I
H
C
F
E
A
D
G
J
B

Symbol	J	B	G	C	F	D	H	A	I	E
Prefix code	110100	110101	11011	0110	0111	1100	010	111	00	10

Example:

Construct the optimal prefix code for the following message and indicate the code.

1. LETTER RECEIVED
2. CALCULUS
3. TAKE CARE
4. PROPOSAL ACEPTED
5. MISSION SUCCESSFUL
6. FALL OF THE WALL
7. MATHEMATICS
8. ENGINEERING

Shortest Paths:-

In **graph** theory, the **shortest path** problem is the problem of finding a **path** between two vertices (or nodes) in a **graph** such that the sum of the weights of its constituent edges is minimized.

Optimization and matching:

- In this section we described a method of constructing a complete binary tree of minimum weight called optimal tree. In this part we consider few more of such optimization Problem.
- Let G be a graph and suppose there is a positive real number associated with each edge of G, then G is called **Weighted graph**.
- A spanning tree whose weight is least is called a minimal **spanning tree**.

Algorithms for Minimal Spanning Tree:

There are several methods of constructing minimal spanning trees. Below we give the working rules of two such Methods. The first of these is due to Kruskal and second is due to Prim

Kruskal Algorithm:

The working rule for the Kruskal’s method (usually called Kruskal’s algorithm) may be stated as follows

- **Step1:** Given a connected, weighted graph G with n vertices, list the edges of G in the order of nondecreasing weights.

- **Step2:** Starting with a smallest weighted edge, proceed sequentially by selecting one edge at a time such that no cycle is formed.

- **Step3:** Stop the process of step2 when (n-1) edges are selected. These (n-1) edges constitute a minimal spanning tree of G.

Remarks

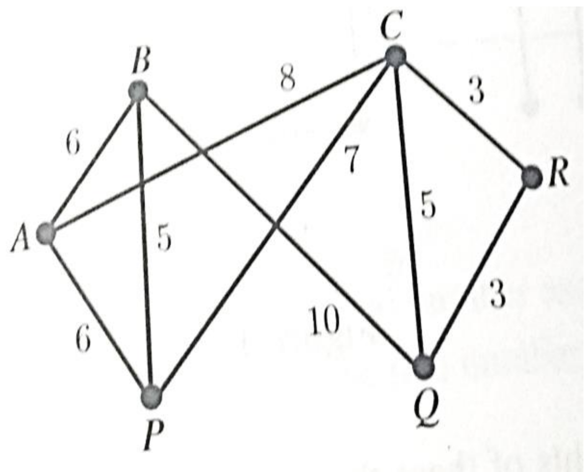
- If two or more edges have the same weight, there will be more than one listing of edges in non-decreasing order of weights. Different listing may yield different minimal spanning trees. As such, Kruskal algorithm does not determine a unique minimal spanning tree.

- The process in step2 is called Greedy Process.

1. **Using the Kruskal's algorithm find a minimal spanning tree of the given weighted graph**

Solution:

We observe that the given graph has 6 vertices; hence a spanning tree there of will have 5 edges (branches).

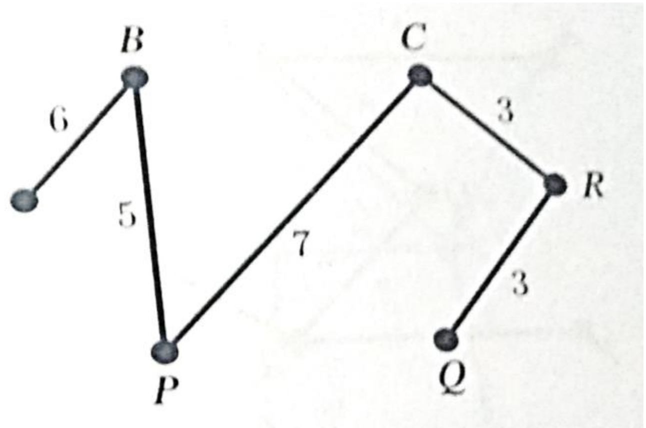


Let us put the graph in a non decreasing order of their weights and successively select 5 edges such a way that no cycle is created. This scheme is summarized in the following table

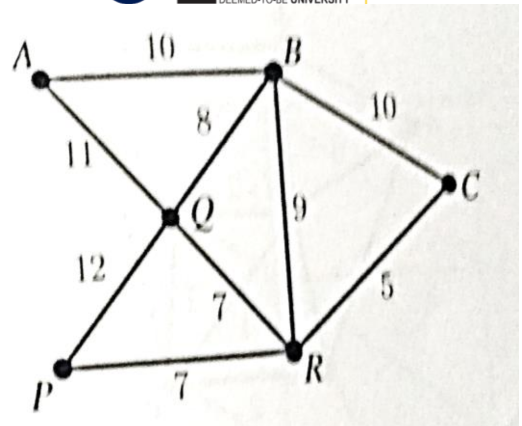
Edge	CR	QR	BP	CQ	AB	AP	CP	AC	BQ
Weight	3	3	5	5	6	6	7	8	10
Select?	YES	YES	YES	NO	YES	NO	YES		

Observe that CQ is not selected because CR and QR have already been selected and the selection of CQ would have created cycle. Further AP is not selected because it would have created cycle along with BP and AB which have been selected. We have stopped the process when exactly 5 edges are selected.

Thus, a minimal spanning tree of the given graph contains the five edges CR, QR, BP, AB, CP. This tree is as shown figure. The weight of this tree is 24 units.



2. Using Kruskal's Algorithm, find a minimal spanning tree for the given weighted graph



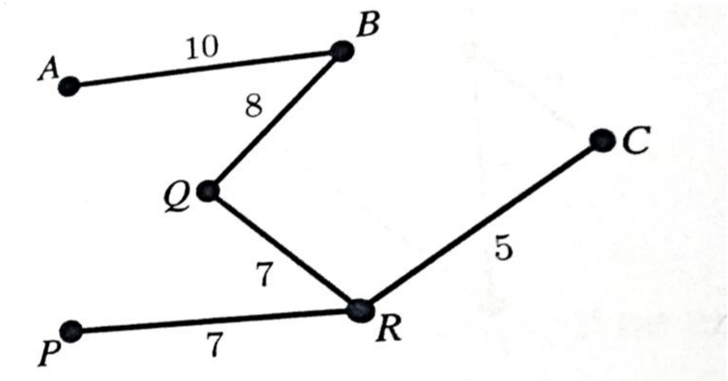
Solution

We observe that the given graph has 6 vertices; hence a spanning tree there of will have 5 edges (branches).

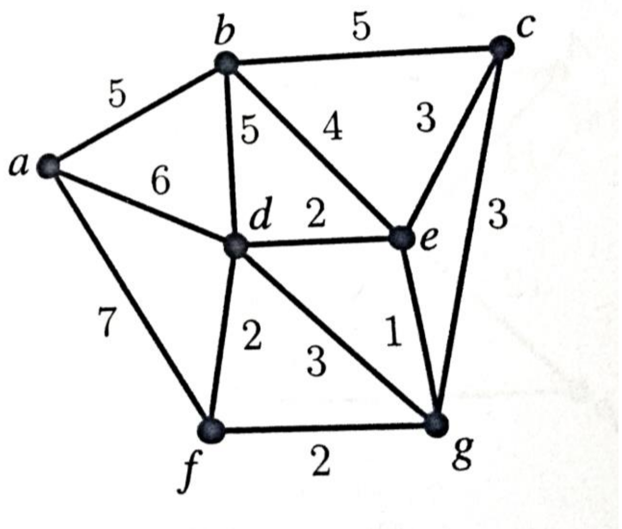
Let us put the graph in a non decreasing order of their weights and successively select 5 edges such a way that no cycle is created. This scheme is summarized in the following table

Edge	CR	PR	QR	BQ	BR	AB	BC	AQ	PQ
Weight	5	7	7	8	9	10	10	11	12
Select?	YES	YES	YES	YES	NO	YES			

Thus, a minimal spanning tree of the given graph contains the five edges CR, PR, QR, BQ, AB. This tree is shown in the figure. The weight of the tree is 37 units.



3. Using Kruskal's Algorithm, find a minimal spanning tree for the given weighted graph.



Prime’s Algorithm:

The working rule for the Prim’s Method(Usually called Prim’s algorithm) may be stated as follows

- **Step1:** Given a connected, weighted graph G with n vertices, Assign n names (say v1,v2,…,vn or A, B, C, and so on) to these vertices, and prepare a nxn table in which the weights of all edges are shown. The entries in the table will be symmetric with respect to the diagonal and no entries appear on the diagonal, indicate the weights of the non existing edges as ∞.

- **Step2:** Start from vertex v1 (or A, as the case may be) and connect it to its nearest neighbour (i.e., to the vertex has the smallest entry) in the v1 row, say vk. Now, consider the edge {v1, vk} and connect it to its closest neighbour (i.e., to a vertex, other than v1 and vk, that has the smallest entry among all entries in v1 and vk rows). Let this be vm.

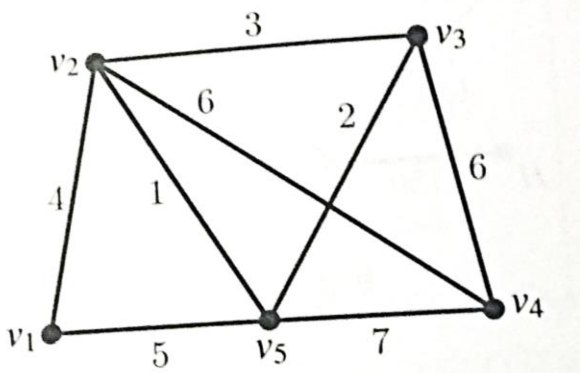
- **Step3:** Start from the vertex vm and repeat the process of step2. stop the process when all the n vertices have been connected by n-1 edges. These n-1 edges constitute a minimal spanning tree.

Remarks

- In the process of connected an edge to its nearest neighbour as explained above, care has to be taken that cycle are not created by the connections.

- Like in the Kruskal’s method, a minimal spanning tree determined by the prims method is not unique.

1. Using Prim algorithm find a minimal spanning tree for the given weighted graph.



Solution

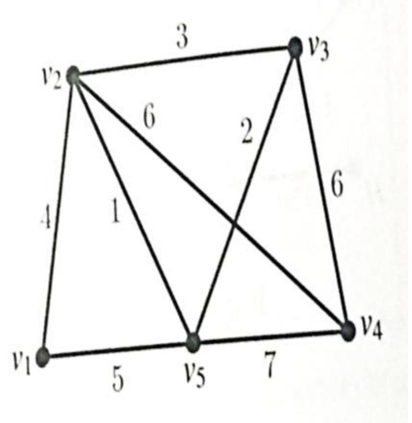
We observe that the graph has 5 vertices. Therefore the minimal spanning tree there of will have edges.

Let us tabulate the weights of the edges between every pair of vertices as shown below

	v1	**v2**	**v3**	**v4**	**v5**
v1	-	4	∞	∞	5
v2	4	-	3	6	1
v3	∞	3	-	6	2
v4	∞	6	6	-	7
v5	5	1	2	7	-

Now let us start with the first row (v1 row) and pick the smallest entry therein. This is 4 which corresponds to the edge {v1, v2}. By examining all the entries in v1- and v2-rows, we find that the vertex other than v1 and v2 which corresponds to smallest entry is v5 (smallest entry being 1). Thus v5 is closest to the edge {v1, v2}. Let us connect v5 to the {v1, v2} at v2 (because {v5, v2} has smaller weight than {v5, v1}).

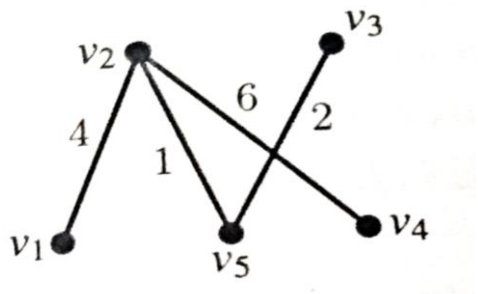
Let us now examine the v5-row and note that small entry is 1 which corresponds to the edge {v5, v2}. By examining all entries in v2- and v5 rows, we find that the vertex other than v2 and v5 which corresponds to the smallest entry v3. thus v3 is closest to the edge {v2, v5}.



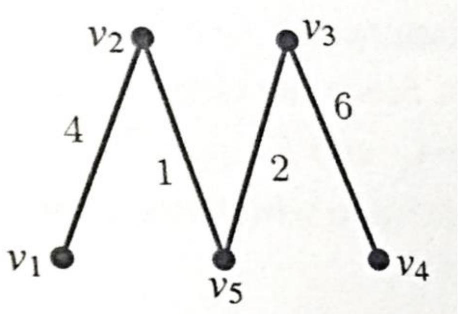
Let us connect v3 to the edge {v2, v5} at v5.

Thus the edges {v1, v2}, {v2, v5}, {v5, v3} belongs to a minimal spanning tree. The vertices left over at this stage is v4 which is joined to v2, v3, and v5 in the given graph. Among the edges that contain v4, the edges {v2, v4} and {v3, v4} have equal minimal weights. Therefore we can include either of those edges in the minimal spanning tree.

Accordingly, the degree {v1, v2}, {v2, v5}, {v5, v3} together with the edge {v2, v4} or the edge {v3, v4} constitute a minimal spanning tree. Thus, for the given graph, there are two minimal spanning trees as shown in the figure (a), (b). The weight of each of these trees is 13 units.

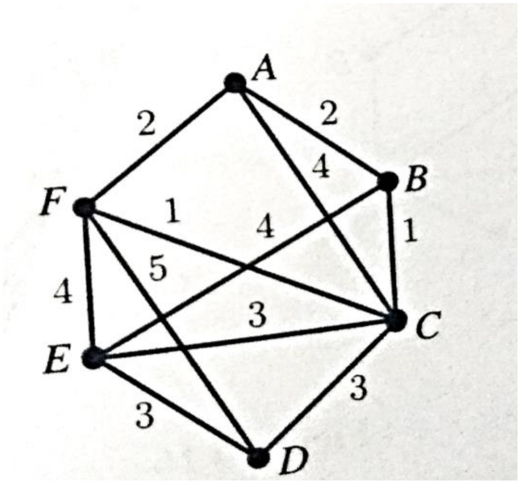


(a)



(b)

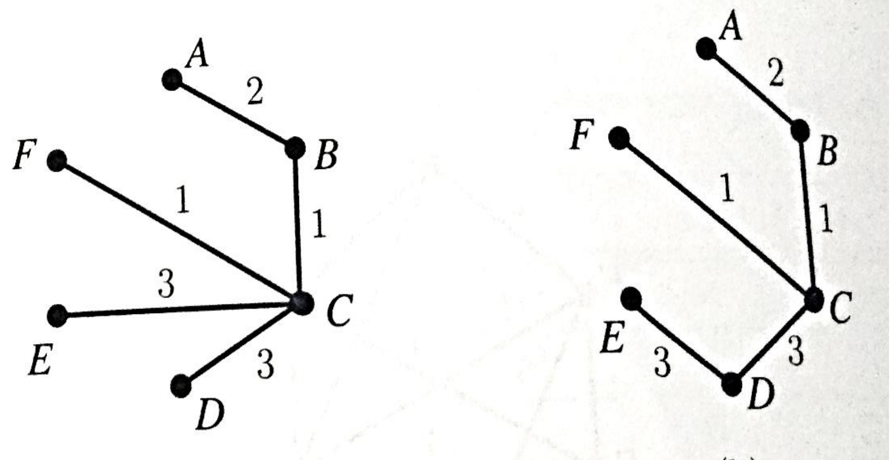
2. Using Prim’s algorithm, find a minimal spanning tree of the weighted graph



Solution

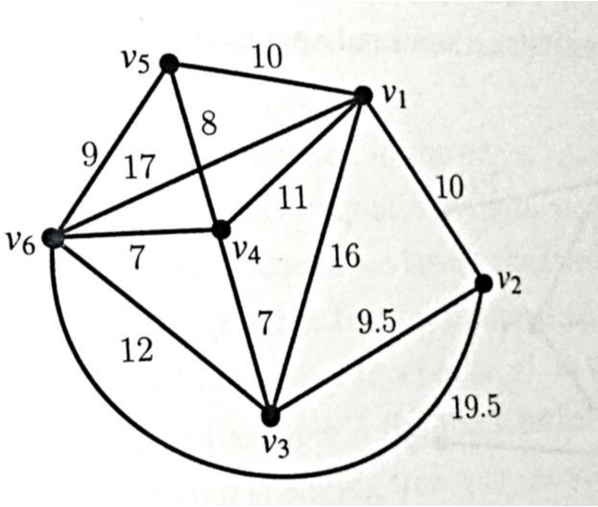
We observe that the graph has 6 vertices. Therefore the minimal spanning tree thereof will have 5 edges. Let us tabulate the weights of the edges between every pair of vertices as shown below

	A	B	C	D	E	F
A	-	2	4	∞	∞	2
B	2	-	1	∞	4	∞
C	4	1	-	3	3	1
D	∞	∞	3	-	3	5
E	∞	4	3	3	-	4
F	2	∞	1	5	4	-



(a) (b)

3. Using Prim's algorithm find the minimal spanning tree for the weighted graph.




Thank you