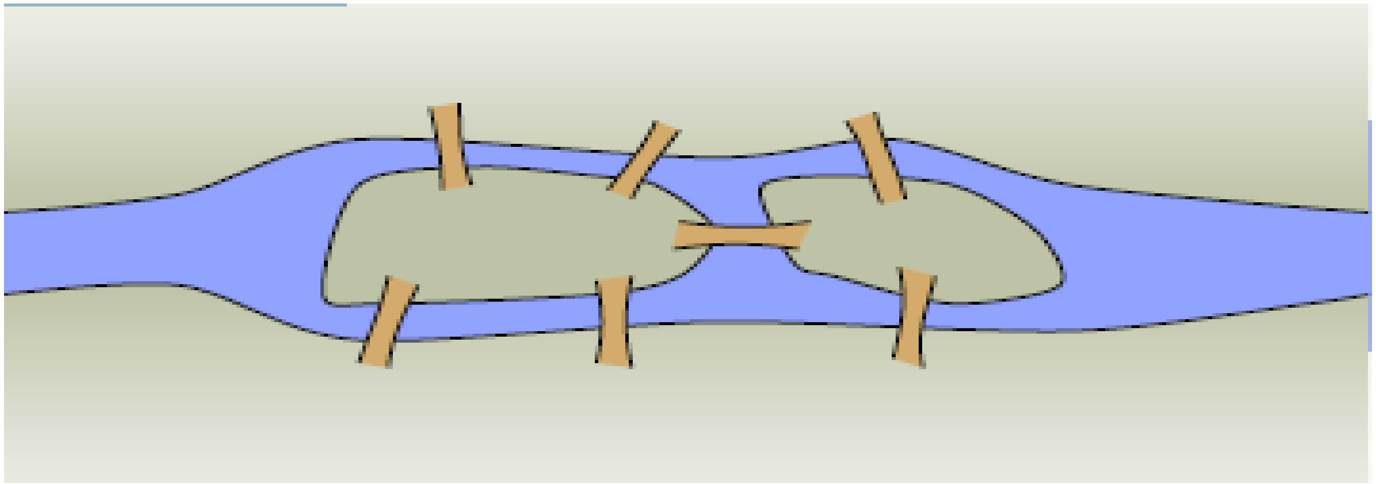
SEMESTER III.
MODULE 5
CO-4
GRAPH THEORY

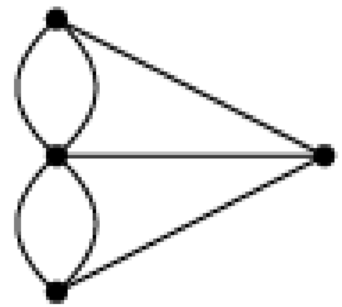
MODULE 5 (CO-4)
UNIT :5.1
INTRODUCTION TO GRAPHS

In the time of Euler, in the town of Konigsberg in Prussia, there was a river containing two islands. The islands were connected to the banks of the river by seven bridges (as seen below). The bridges were very beautiful , so townspeople would spend time walking over the bridges , on their days off



As time passed, a question arose: was it possible to plan a walk so that you cross each bridge once and only once with same stating & ending point?
Euler was able to answer this question.
Are you?
The problem above, known as the *Seven Bridges of Konigsberg*, is the problem that originally inspired graph theory.
Graph Theory is first studied by the super famous mathematician Leonhard Euler in 1735.

Consider a "different" problem:



Here are four dots connected by some lines.

Is it possible to trace over each line once and only once (without lifting up your pencil, with same starting and ending dot)?

Is there connection between these two problems ?

There is an obvious connection between these two problems

Definition: Graph

A graph is a ordered pair G = (V,E)

Where *V* is a set of vertices and

E is a set of edges (2-element subsets of *V)*.

Note :

A vertex is also called as node or point .

An edge is also called as curve or line .

Example :

G=(V={a, b, c, d}, E={{a, b},{a, c},{b, c},{b, d},{c, d}})

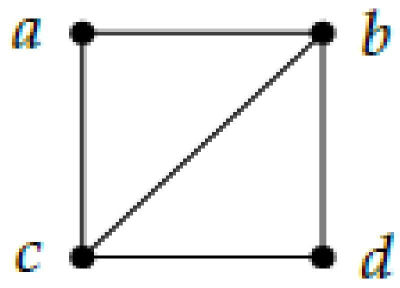
Here we have a graph G with V={a, b, c, d} and E={{a, b}, {a, c}, {b, c}, {b, d}, {c, d}}

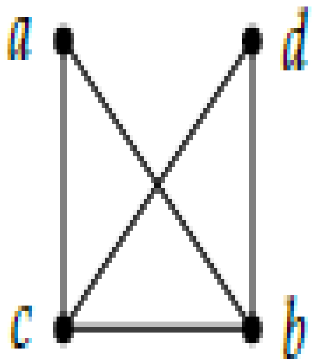
Looking at V ,E it is difficult to process.

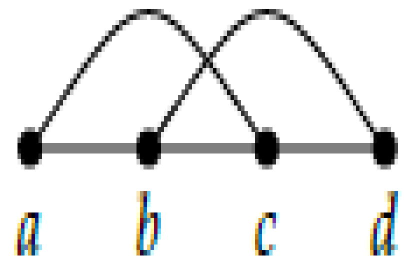
That is why we often draw a representation of these sets.

We put a dot down for each vertex, and connect two dots with a line precisely when those two vertices are one of the 2-element subsets in our set of edges.

Thus the graph G can be drawn as:







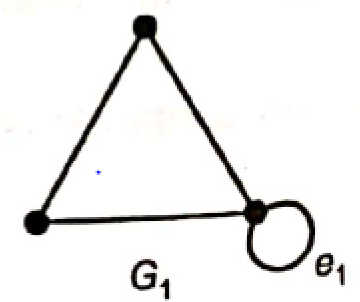
Definition : Loop

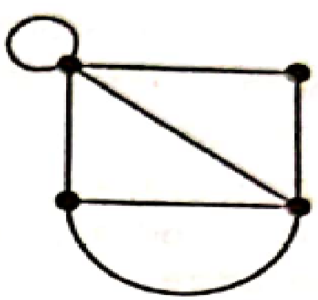
If an edge starts and ends at same vertex then it is called as **Loop**

Definition :Parallel edges/Multiple edges

If two edges have same starting vertex and same end vertex then they are called as **Parallel edges**

Example :





Definitions

Simple graph: A graph having no loops and no parallel edges is called as Simple graph

Degree of a vertex: The number of edges connected to a vertex ‘a’ is called as degree of a vertex ‘a’. It is denoted by d(a).

Null Graph – Graph with No edges

Trivial Graph – Graph with single vertex

Multigraph – Graph with multiple edges

Diagraph – Graph with directed edges

Isolated vertex – Vertex with deg zero

Pendent vertex – Vertex with deg one

Note: for a loop at v, deg is counted twice at v.

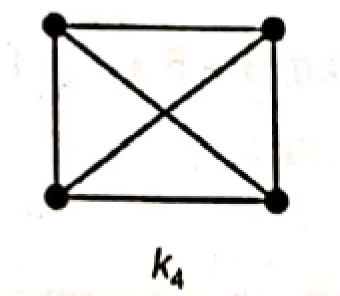
Adjacent: Two vertices are **adjacent** if they are connected by an edge.

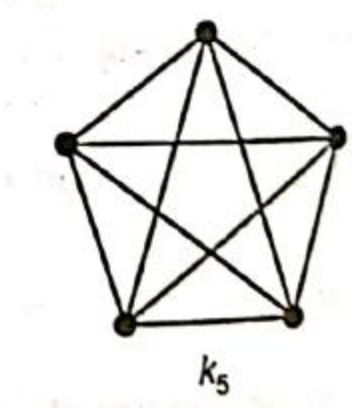
Two edges are **adjacent** if they share a vertex.

Complete graph:

A simple graph in which every pair of vertices is adjacent is called as Complete graph .

The complete graph on *n* vertices is dented by K_n





Handshake Lemma

In any graph, the sum of the degrees of vertices in the graph is always twice the number of edges.

$$\sum_{v \in V} d(v) = 2e.$$

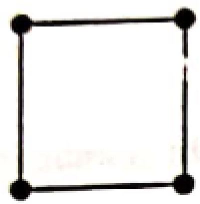
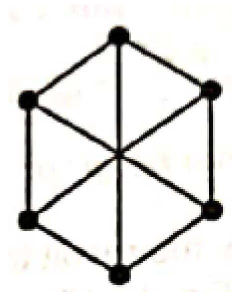
Proposition: *In any graph, the number of vertices with odd degree must be even*

In a simple Graph:

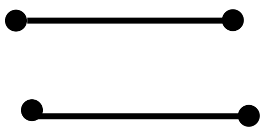
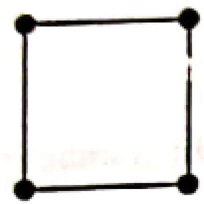
Maximum degree of any vertex is (n-1)
Why?

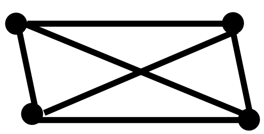
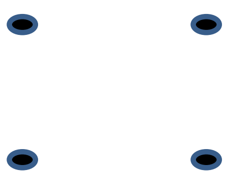
Simple graph with n vertices has atmost n(n-1)/2 edges
Why?

Definition: If a graph in which degree of every vertex is r then graph is called as **r-regular graph**

2-regular -  3-regular 

For Given 4 vertices

1 regular  2 regular 

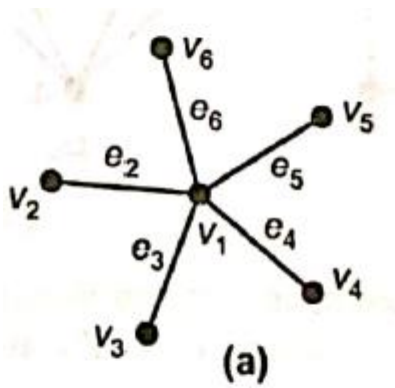
3 regular  0 regular 

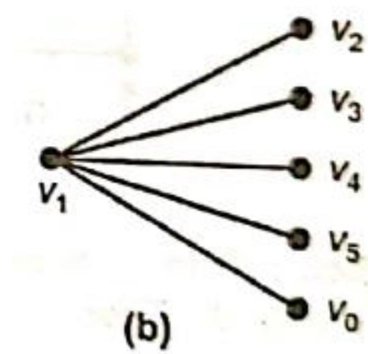
Note That,

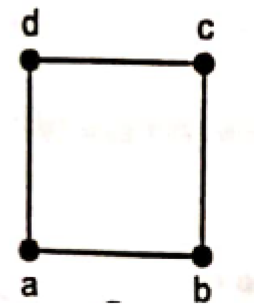
1. If G is r-regular graph with n vertices and m edges then $m = \frac{nr}{2}$ (Check this for examples)
2. If r and n are odd no. then there can not be an r-regular graph of n vertices. (Check!!)

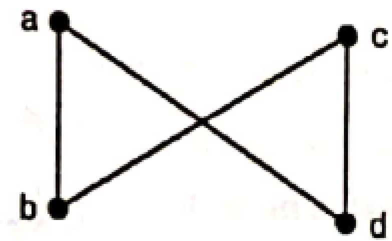
Definition: A graph G = (V,E) is called as **bipartite graph**
If (i) V can be expressed as union of 2 disjoint sets U,W
(ii) Every edge in E has one vertex in U and one vertex in W
Where U,W are called partition of V

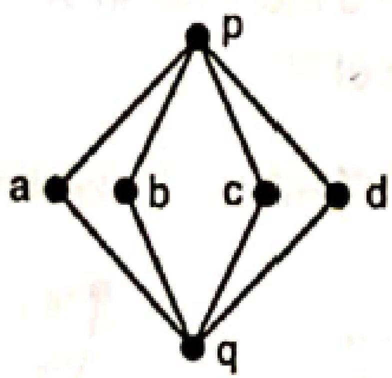
Example :

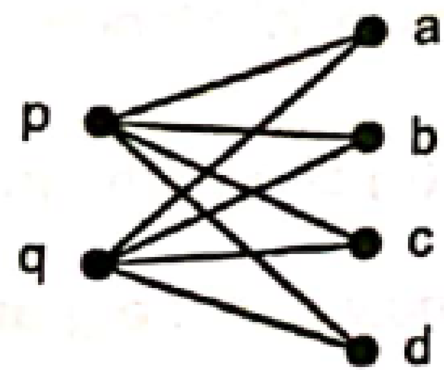
1) 

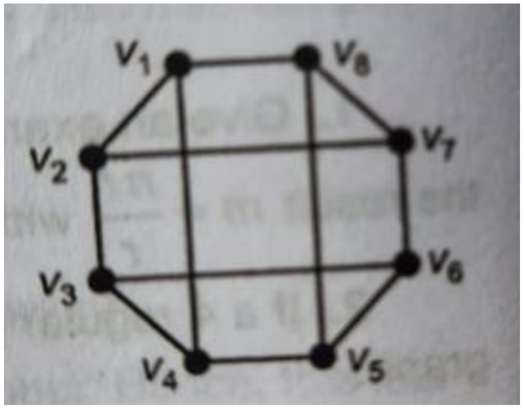
Rearrange 

2) 

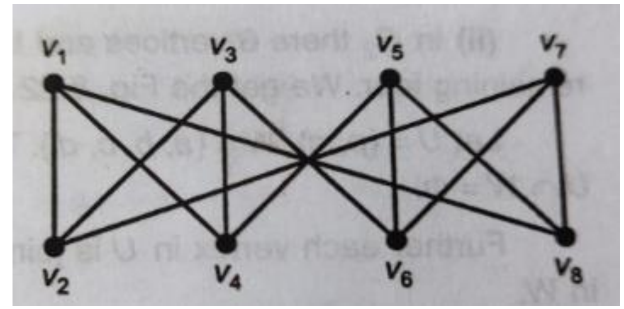
Rearrange 

3) 

Rearrange 

4) 

Rearrange

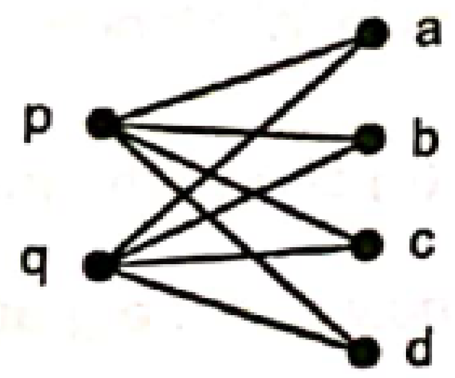


Definition: complete bipartite graph

A bipartite graph G = (V,E) is called as complete bipartite graph if each vertex in *U* is adjacent to all the vertices in *W*

complete bipartite graph is denoted by $K_{m,n}$

Where |U|=m ,|V|=n



Which of the bipartite graphs we have seen are complete bipartite?

Definition : Isomorphic graphs

Let G1 = (V1,E1) & G2 = (V2,E2) be graphs

If ∃ (there exist) bijective function (isomorphism)

$f: V1 \rightarrow V2$ with {a,b} $\in$ E1 iff {f(a),f(b)} $\in$ E2

({a,b} is an edge of G1 iff {f(a),f(b)} is an edge of G2)

Then G1 &G2 are called isomorphic graphs

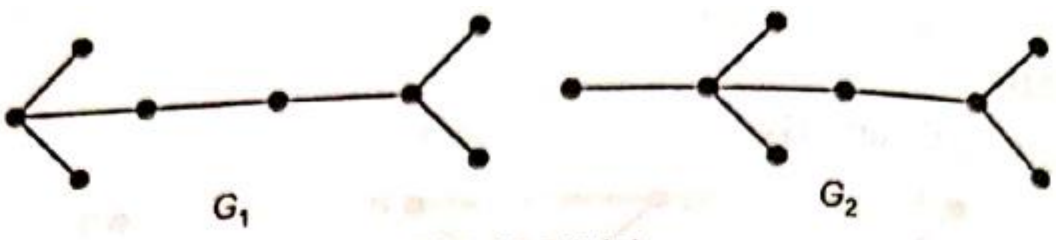
They are denoted by G1 $\cong$G2

Note: For isomorphic graphs, G1 and G2

1. They must have same number of vertices
2. They must have same number of edges
3. They must have same degree for corresponding vertices. i.e. If $d(a) = 3$ then $d(f(a)) = 3$
4. The isomorphism must preserve the vertex and edge adjacency.

If any one of above conditions is not satisfied then graphs *G1 and G2* are not isomorphic graphs

Example: Check whether graphs below are isomorphic ?



G1 and G2 are not isomorphic graphs

In graph *G1* there are 4 vertices of degree 1

In graph *G2* there are 5 vertices of degree 1

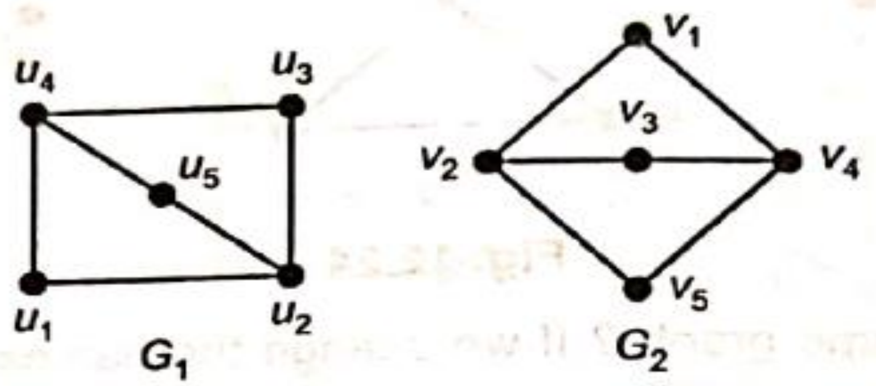
OR

G2 contains one vertex of deg 4 whereas G1 does not have any vertex of deg 4

OR

G1 has 2 vertices of deg 2 whereas G2 has only one.

Example :Check whether following graphs are isomorphic ?



The two graphs G_1, G_2 have five vertices and six edges each. Further, i there are three vertices of degree two and two vertices of degree three.

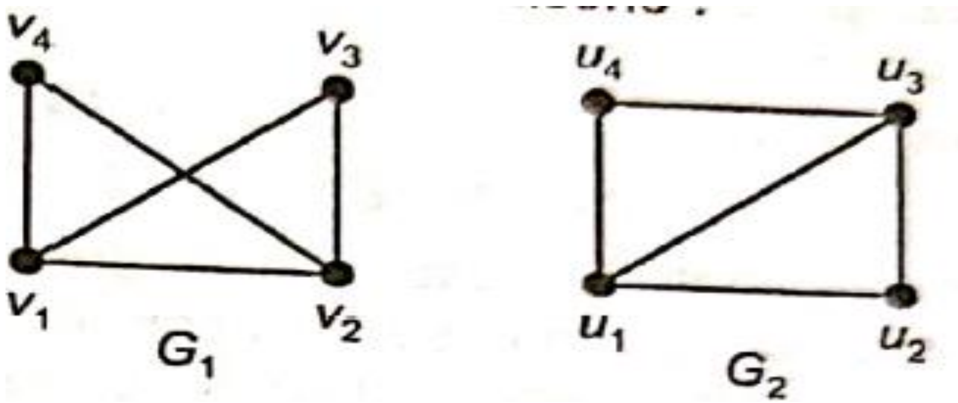
We can define one-to-one correspondence f as follows :

$$u_1 \to v_5,\ u_2 \to v_4,\ u_3 \to v_1,\ u_4 \to v_2,\ u_5 \to v_3.$$

This correspondence preserves the adjacency and incidence relationship

$\therefore$ The graphs are isomorphic.

Example :Check whether following graphs are isomorphic ?



G1 and G2 are isomorphic graphs with 4 vertices and 5 edges. There are 2 vertices of deg 2 and 2 vertices of deg 3 in both graphs. One of the isomorphism is given by,

$$v_1 \to u_1,\ v_3 \to u_2,\ v_2 \to u_3,\ v_4 \to u.$$

Check that, it preserves the adjacency.

Definitions

Walk

When we transverse the graph as we want, it is called as walk.

In a walk, edges & vertices can be repeated.

Trail

A walk in which no edge is repeated but vertices can be repeated is called as trail.

Circuit

A trail in which starting & ending vertex is same is called as circuit OR

A walk in which no edge is repeated but vertices can be repeated with same starting & ending vertex is called as circuit.

Definitions

Path

A walk in which edges & vertices are not repeated.

Cycle

A path in which starting & ending vertex is same is called as cycle.

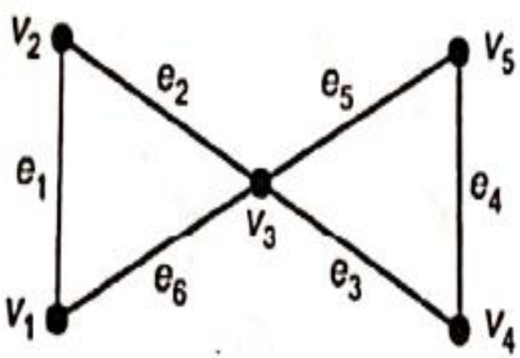
OR

A walk in which no edge is repeated & no vertex is repeated except starting & ending vertex is called as cycle.

A walk in which no edge is repeated & no vertex is repeated except starting & ending vertex is called as cycle

A walk in which no edge is repeated but vertices can be repeated with same starting & ending vertex is called as circuit.

Example :



$v_1\ e_1 v_2 e_2\ v_3 e_3\ v_4 e_4\ v_5 e_5\ v_3 e_6 v_1$ **is a circuit (not cycle)**

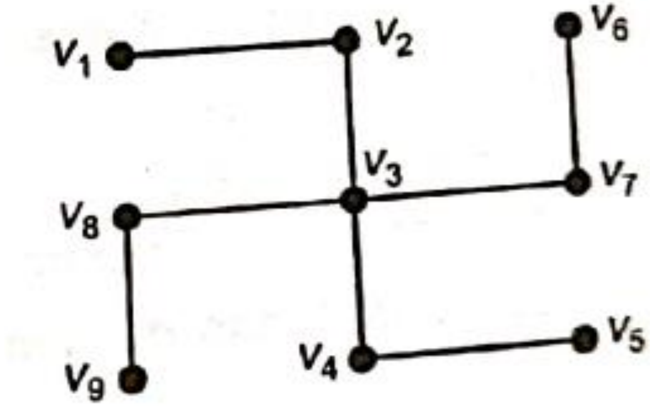
$v_1\ e_1 v_2 e_2\ v_3 e_6 v_1$ **is a cycle (or circuit)**

Definition : Path

path : A sequence of vertices $[v_0, v_1, v_2, \ldots v_l]$ is a *path from v_0 to v_l of length l*

in G if $v_{i-1}v_i \in E$ for i=1,2, …,l.

Example



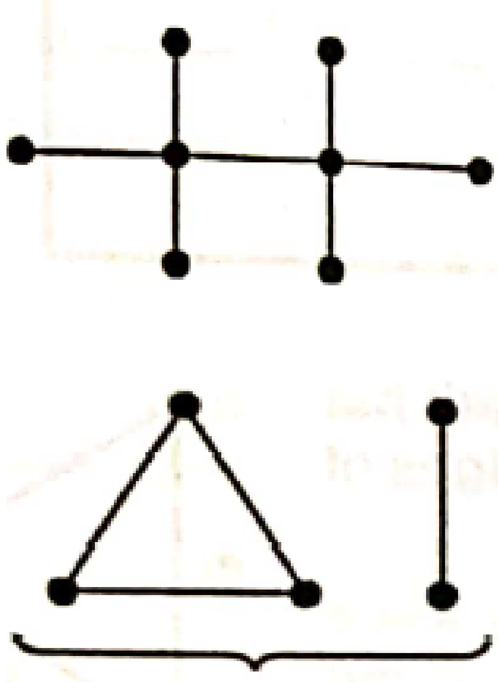
$v_1\ v_2\ v_3\ v_4\ v_5$ is a path of length 4

$v_1\ v_2\ v_3\ v_7$ is a path of length 3

Definition : Connected graph

A graph is called connected graph if there is a path from any vertex to any other vertex

Example :



Edge Connectivity

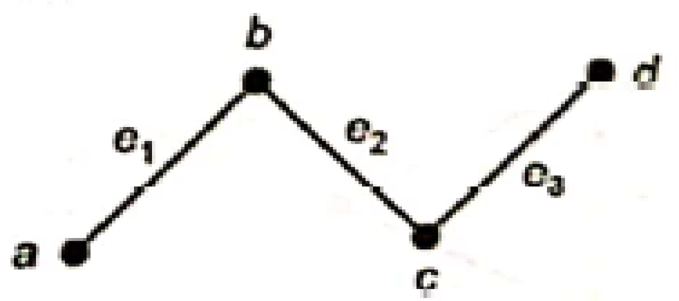
Vertex Connectivity

Applications in Military, traffic and communications??

Definition : Eulerian Path/ Circuit/ Graph

An Eulerian path is a path through the graph in which **every** edge of a graph G appears in the path and it appears **exactly once** in a path.

Example :

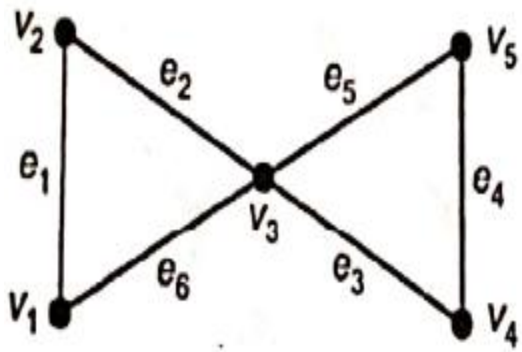


$ae_1\ be_2$c e_3d is an Eulerian path

A circuit in which **every** edge of a graph G appears **exactly once** is called an **Eulerian circuit**.

A Graph which has an Eulerian circuit is called an **Eulerian Graph**.

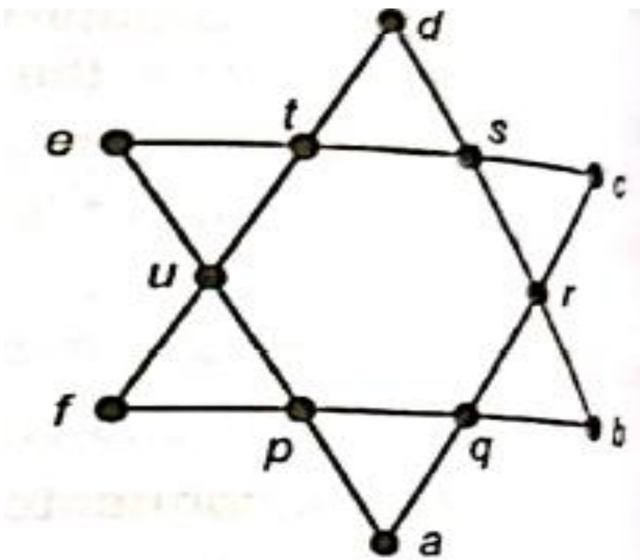
Example :

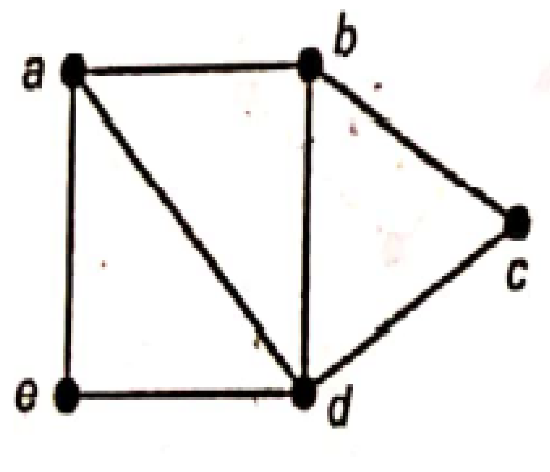


$v_1\ e_1 v_2 e_2\ v_3 e_3\ v_4 e_4\ v_5 e_5\ v_3 e_6 v_1$ is an Eulerian circuit.
Note, There is No Eulerian cycle in the Graph due to V_3

Theorem 1 : G is an Eulerian graph if and only if all vertices of G are of even degree

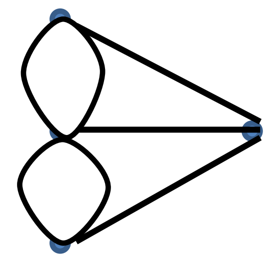
Theorem 2 : G is a connected graph having exactly 2 vertices (say u,v) of odd degree then G contains Eulerian trail from u to v .

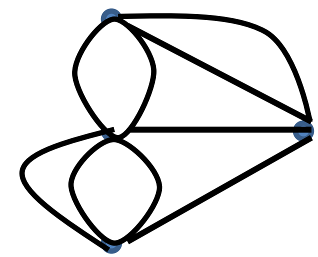




Example : Is Konigsberg seven bridge problem solvable or not? What is the minimum number of bridges to be built in order to make it solvable. (Find all possible solutions for them)

ANS. Konigsberg problem is isomorphic to following graph. The problem is solvable if we could find an Eulerian circuit in this graph.





But this graph is not Eulerian.
(Check the degrees of vertices)
Therefore Seven bridge problem is not solvable.
We can make it solvable by making degrees of each vertex even. (One solution is depicted. How may different ways are possible?)

Definition : Hamiltonian Graph

A connected graph G is called **Hamiltonian graph** if there is a cycle which includes every vertex of G and that cycle is called **Hamiltonian cycle**.

A connected graph G is called **Semi-Hamiltonian graph** if there is a path which includes every vertex of G and that path is called **Hamiltonian path**.

Is Hamiltonian $\Rightarrow$ Semi-Hamiltonian?

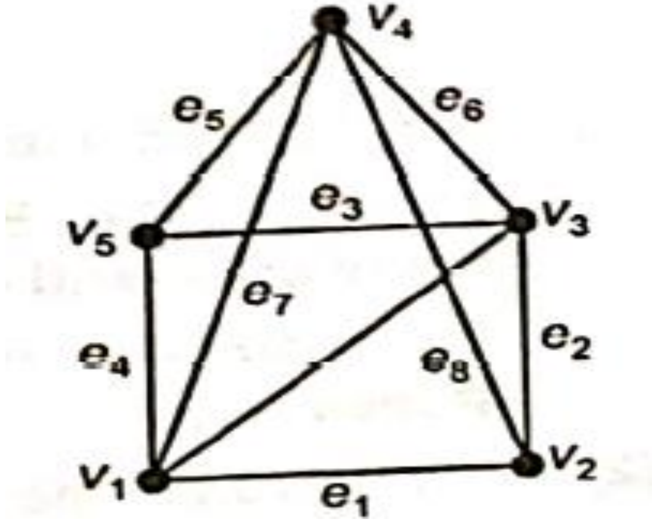
Do all edges are traversed. ?

In Hamiltonian cycle, what is deg of each vertex ?

Is Complete graph Hamiltonian?

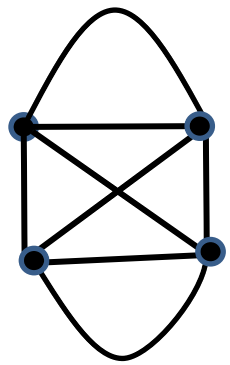
Is Eulerian $\Rightarrow$ Hamiltonian? Or Vice-versa?

Example :

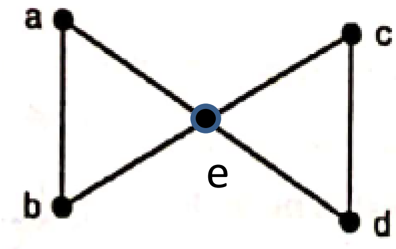


$v_1\ e_1 v_2 e_2\ v_3 e_6\ v_4 e_5\ v_5 e_4\ v_1$ is Hamiltonian cycle

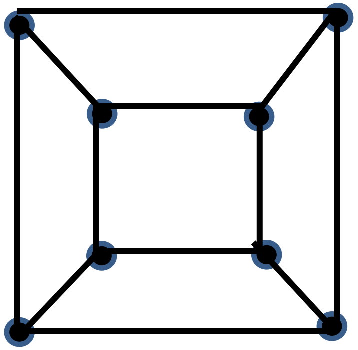
Examples: Check for Eulerian and Hamiltonian



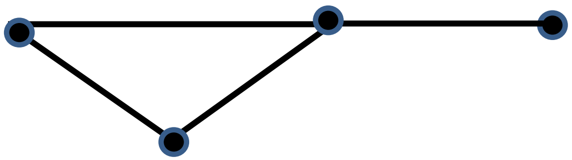
E and H



E but NH



NE but H



NE and NH

Theorem 2 : If G is a simple connected graph with n vertices, where deg (x) ≥ (n/2) for each vertex, then the graph G is Hamiltonian graph.
Example: K5, check previous examples

Note : converse is not true.

Travelling Salesperson Problem (T.S.P.)

One of the applications of Hamiltonian cycle is Travelling Salesperson Problem (T.S.P.)

A salesperson is required to start from the headquarter V and visit the retail shops located at a,b,c,d,……..e then come back to the headquarter at V . If the distance from any place to any other place is known, then what should be the order of the places in which salesperson should visit the places so that distance travelled is minimum. This is called as Travelling Salesperson Problem

Travelling Salesperson Problem is solved by the method called as nearest neighbour method, procedure is as follows.

1.Start from vertex V which denotes the headquarter. From V go to the vertex which has minimum distance from V.

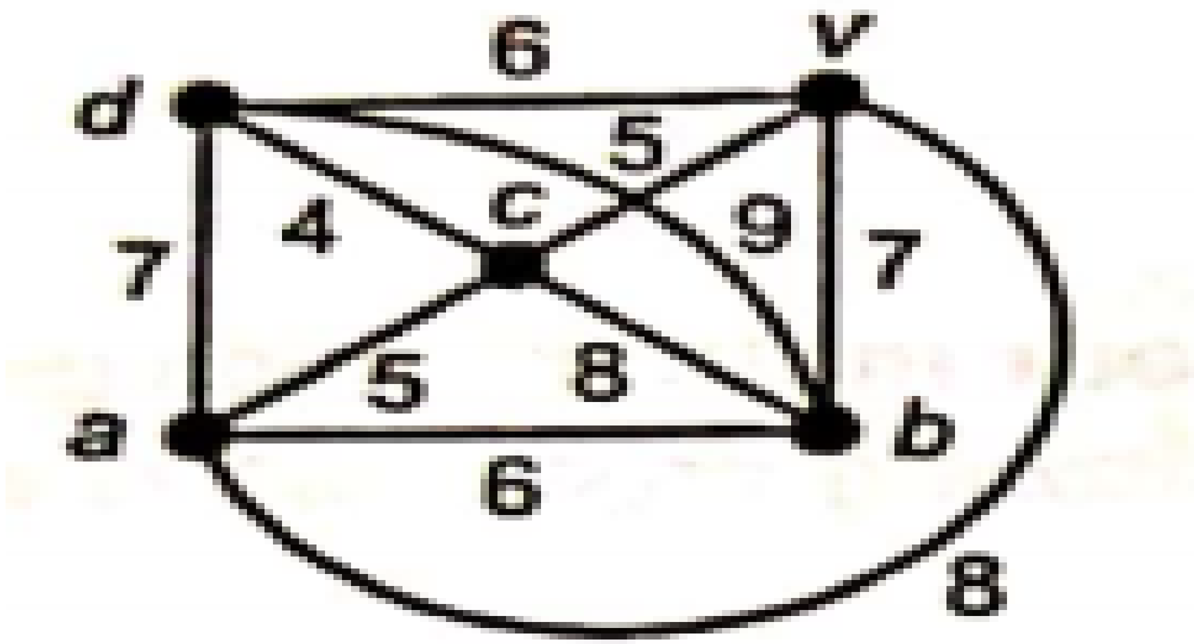
2. If V_1 is such vertex then join V to V_1 by an edge. From V_1 go to the vertex which has minimum distance from V_1.

3. If V_2 is such vertex then join V_1 to V_2 by an edge.

4.In this way complete Hamiltonian cycle, joining the last vertex to V back.

Example :

Solve Travelling salesperson problem for the given graph with starting vertex as V and hence find the length of a required Hamiltonian cycle.



Answer :

1.Vertex V is adjacent with all vertices a,b,c,d ,but distance between V & c is minimum . To construct Hamiltonian cycle, consider first edge Vc.

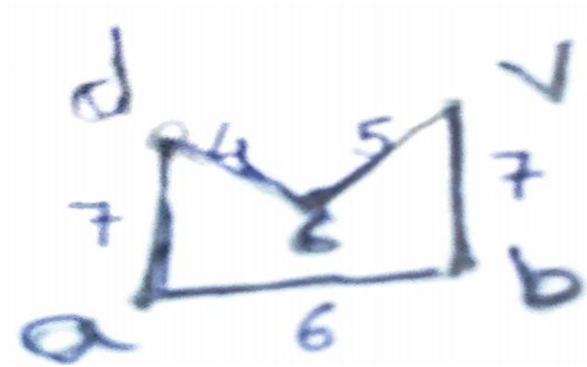
2.Vertex c is adjacent with vertices a, b, d but distance between c & d is minimum. To construct Hamiltonian cycle, consider second edge cd.

3.Vertex d is adjacent with vertices a,b ,but distance between d & a is minimum . To construct Hamiltonian cycle, consider third edge da.

4.Vertex a is adjacent with remaining vertex b. To construct Hamiltonian cycle ,consider next edge ab

5.To complete Hamiltonian cycle , joining the last vertex b to V .

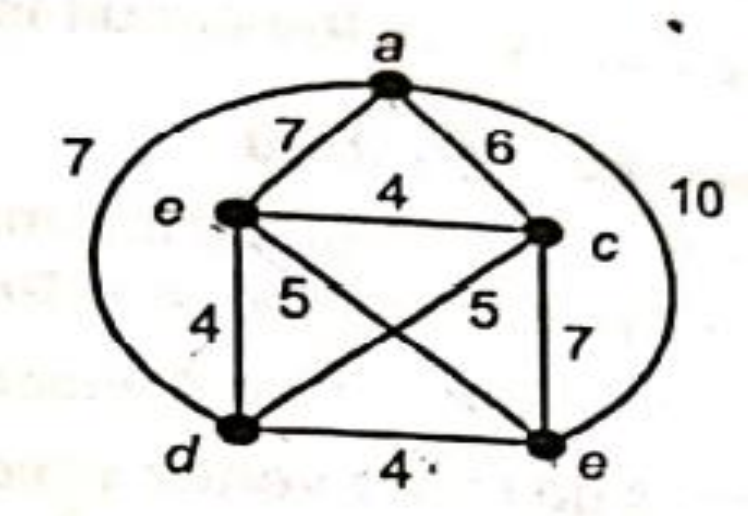
We get Hamiltonian cycle as follows



The length of a required Hamiltonian cycle V-c-d-a-b-V is 5+4 +7+6+7=29 units

Example :

Find the minimum Hamiltonian cycle for the given graph with starting vertex (i) as a and (ii) as d hence find the length of a required Hamiltonian cycle.



Planar Graph

Planar Graph: If it is possible to draw the diagram of the given graph in such away that no 2 edges intersects except at vertex is called as Planar Graph

Plane Graph: The diagram of a graph drawn in such away that no 2 edges intersects except at vertex is called as Plane Graph

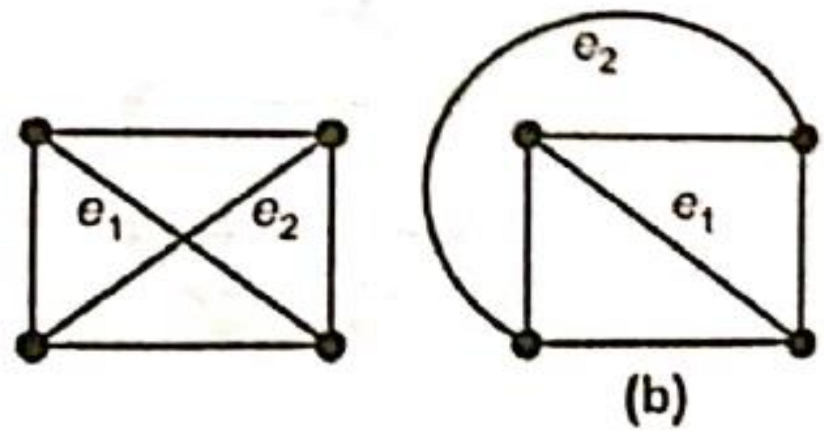
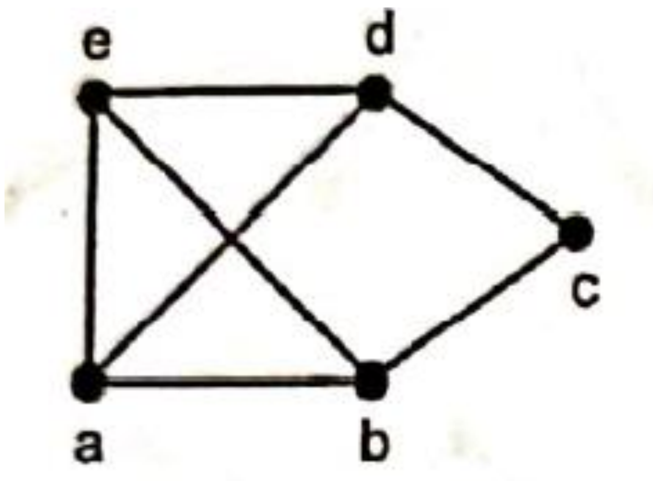
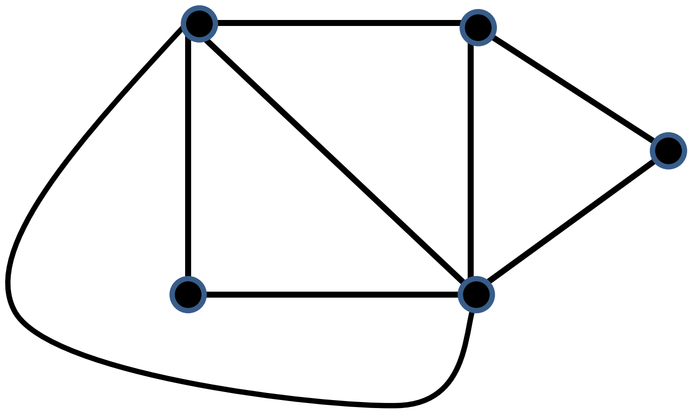
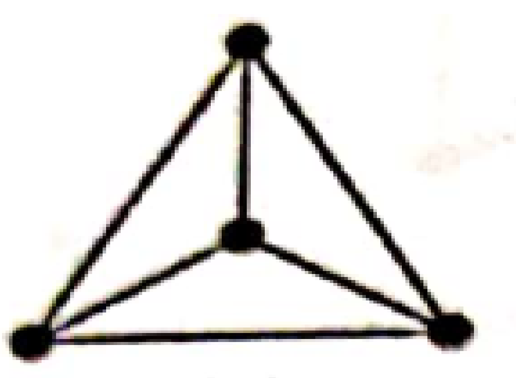


Fig. 12.21 : (a) Planar Graph,
(b) Plane Graph.

Example : **Planar Graph & Plane Graph**



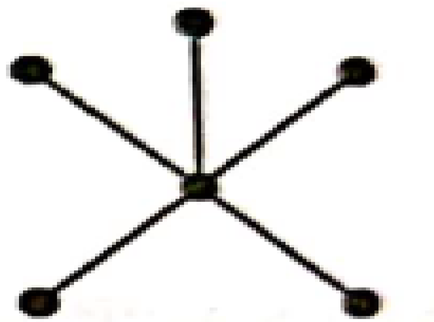


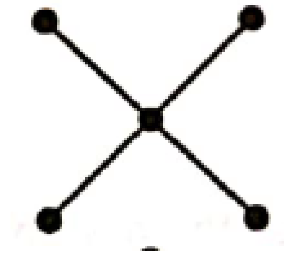


Definition :Tree

A connected graph which does not have cycle (or circuit) is called as tree .

Example:





1. Tree is simple connected graph with No cycles.
2. Tree is plane graph.
3. It is always Bipartite graph.
4. Maximally Non-cycle, minimally connected; Adding edge will get cycle, removing edge will disconnect it.

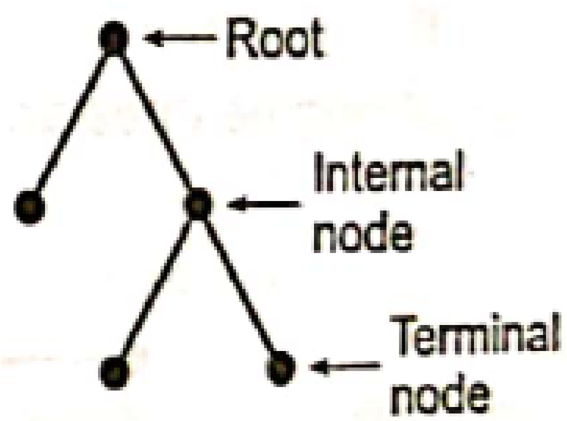
Terminal node

A vertex of degree 1 in tree is called as Terminal node or leaf node.

Internal node

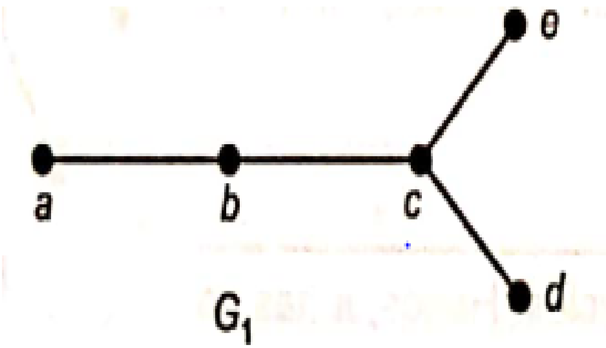
A vertex of degree grater than 1 in tree is called as internal node or branch node .

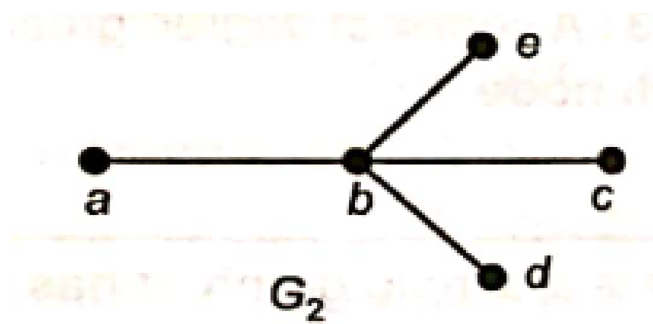
Example :



Example :

Are following trees isomorphic ?





G1 and G2 are not isomorphic trees, Since graph *G2* contains only one vertex of degree 4 and G1 does not have vertex of degree 4

Theorem 1 :Graph is tree if and only if there is one & only one path between every pair of vertices

Theorem 2 : A tree with n vertices has (n-1) edges

Theorem 3 : A connected graph with n vertices having (n-1) edges is tree

Example: A tree T has m vertices of degree one, two vertices of degree two, three vertices of degree four and four vertices of degree three then find m .

$$n = m + 2 + 3 + 4 = m + 9 \qquad \text{................ (1)}$$

Further, the sum of degrees in T

$$= 1 \times m + 2 \times 2 + 3 \times 4 + 4 \times 3$$

$$= m + 28 \qquad \text{................ (2)}$$

i.e., $\sum d(v_i) = m + 28$

Let the tree T have k number of edges.

Since T is a tree with n vertices, $k = n - 1$ (See Theorem 3 above).

But the sum of degrees of all vertices of a graph is equal to twice the number of edges.

$\therefore \quad \sum d(v_i) = 2k = 2(n-1)$

Now, using (1),

$$\sum d(v_i) = 2n - 2 = 2(m + 9) - 2$$

$\therefore \quad \sum d(v_i) = 2m + 16 \qquad \text{................ (3)}$

From (2) and (3), we get

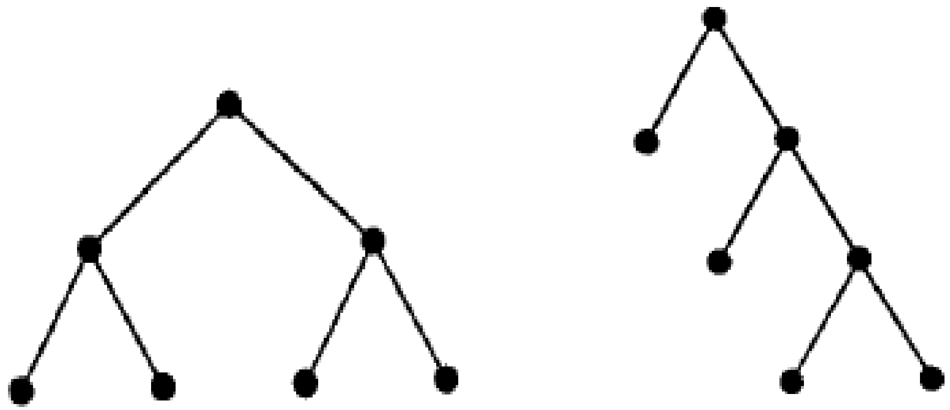
$$m + 28 = 2m + 16 \qquad \therefore \quad m = 12.$$

$\therefore$ There are twelve vertices of degree one.

Definition :

A tree having exactly one vertex of degree two, remaining vertices of degree one or three is called as binary tree.

Example :



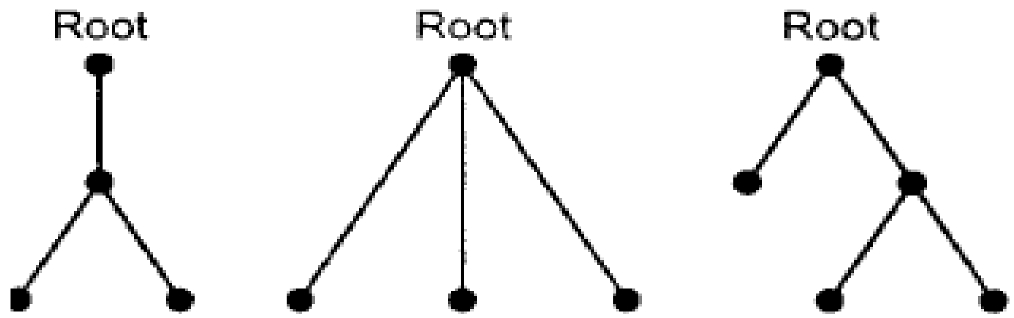
Definition :

A tree in which exactly one vertex is distinguishable from all other vertices is called a **rooted tree**.

The particular distinguishable vertex is called the **root** of the tree.

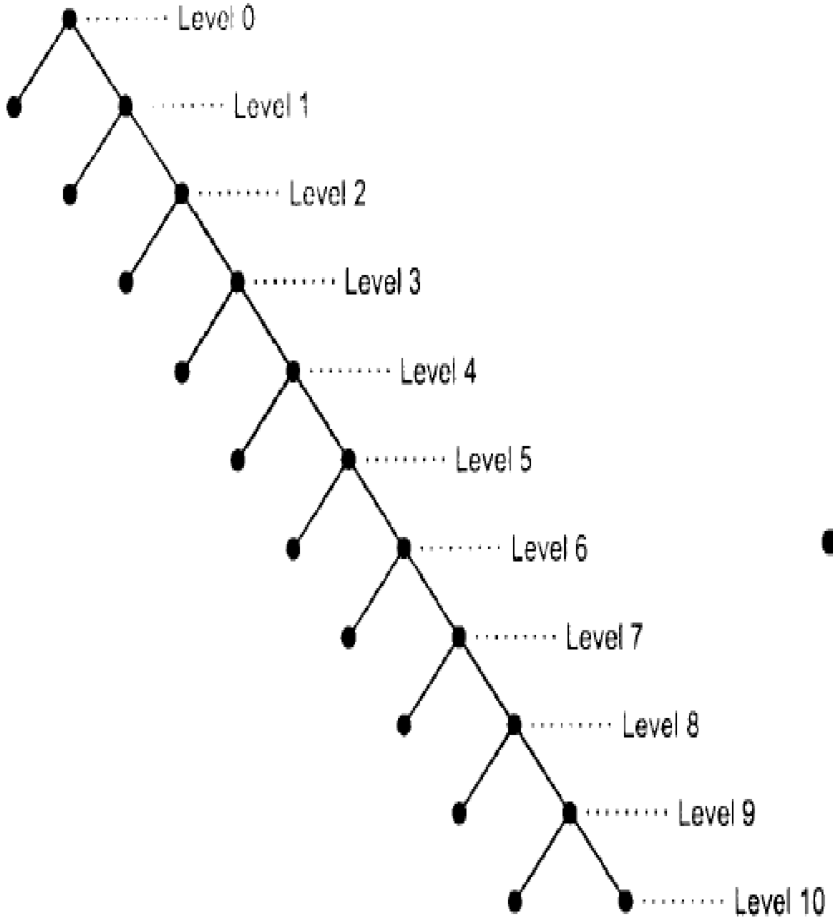
Example

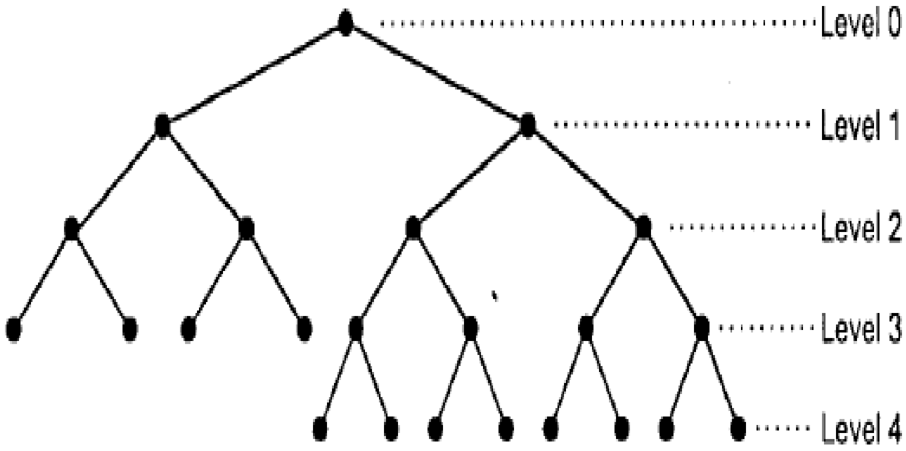
:



Graph:

1) Level of a vertex
2) Height of a Tree





Full binary tree: If in binary tree every internal node has exactly 2 branches it is called as full binary tree.

Pendant : In binary tree vertices of degree 1 are called as pendant vertices.

Non-pendant: In binary tree vertices of degree 3 & 2 are called as non- pendant vertices.

Property 1 : The number of vertices in binary tree is always odd.

Property 2 : The number of pendant vertices in binary tree with n vertices is (n+1)/2

Property 3 : The number of non-pendant vertices in binary tree with n vertices is [(n+1)/2]-1 .

Property 4 : The maximum height of binary tree with n vertices is (n-1)/2 and minimum height is smallest integer greater than or equal to $log_2(n+1)$-1

Let ceil(x) be the smallest integer greater than or equal to x, it is denoted by ⌈x⌉ e.g. ⌈3.65⌉=4 , ⌈0.35⌉=1

Example : Find maximum, minimum height of binary tree with 21 vertices

answer

$$\text{Maximum height} = \frac{21-1}{2} = 10$$

Now, $\log_2(21+1) - 1 = \log_2 22 - 1$

$$= \frac{\log_{10} 22}{\log_{10} 2} - 1 = \frac{1.3424}{0.3010} - 1$$

$$= 4.4598 - 1 = 3.4598$$

$\therefore$ Minimum height = Smallest integer greater than 3·4598 = 4

Example : Find maximum ,minimum height of binary tree with11 vertices

Prefix Code

sequence

An ordered set is called as sequence

010 is a sequence using 0&1 of length3

01110 is a sequence using 0&1 of length5

Example :

C={00,001,01,010,11,111}

In the set C sequence 00 is prefix in (appears before) the sequence 001

A = {000,001,01,10,110,111}

In the set A no sequence is prefix in (occurs before) any other sequence

B ={01,10,001,110,1111}

In the set B no sequence is prefix in any other sequence

Binary prefix code:

A set of sequences formed using digits 0 & 1 is called as a binary prefix code if no sequence is prefix of any other sequence in the set

Example :

A = {000,001,01,10,110,111}

In the set A no sequence is prefix in any other sequence

B ={01,10,001,110,1111}

In the set B no sequence is prefix in any other sequence

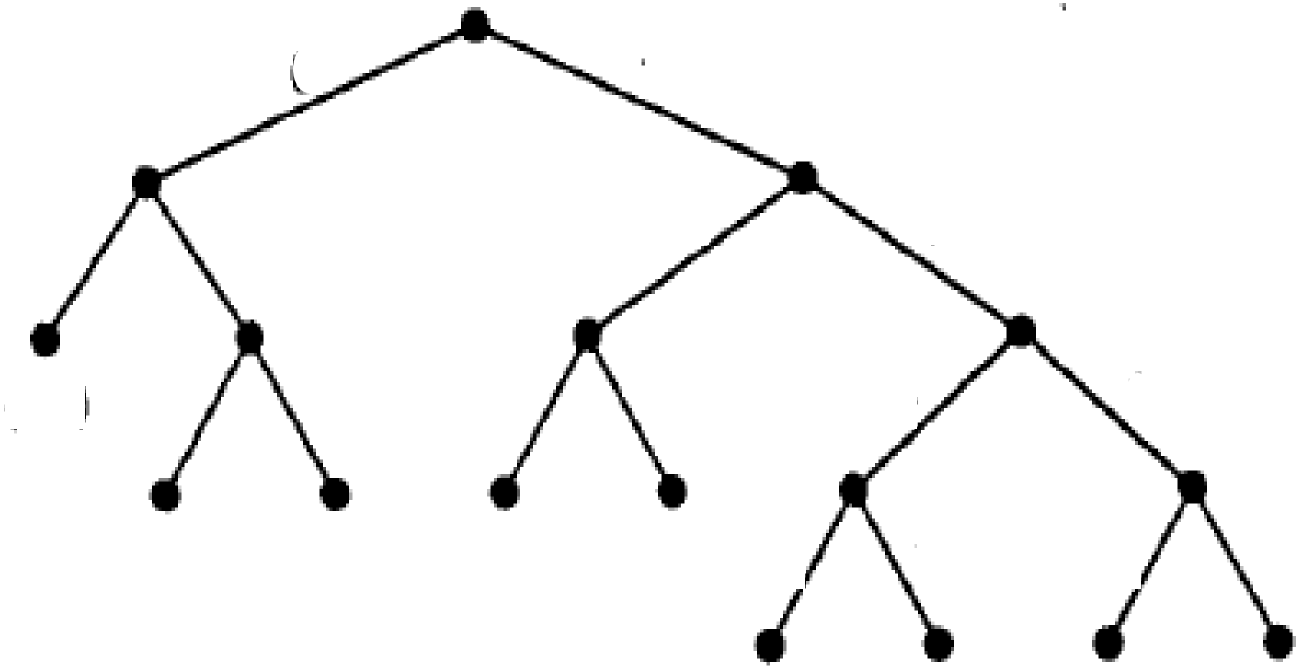
Is A and B are Binary prefix code?

REMARK

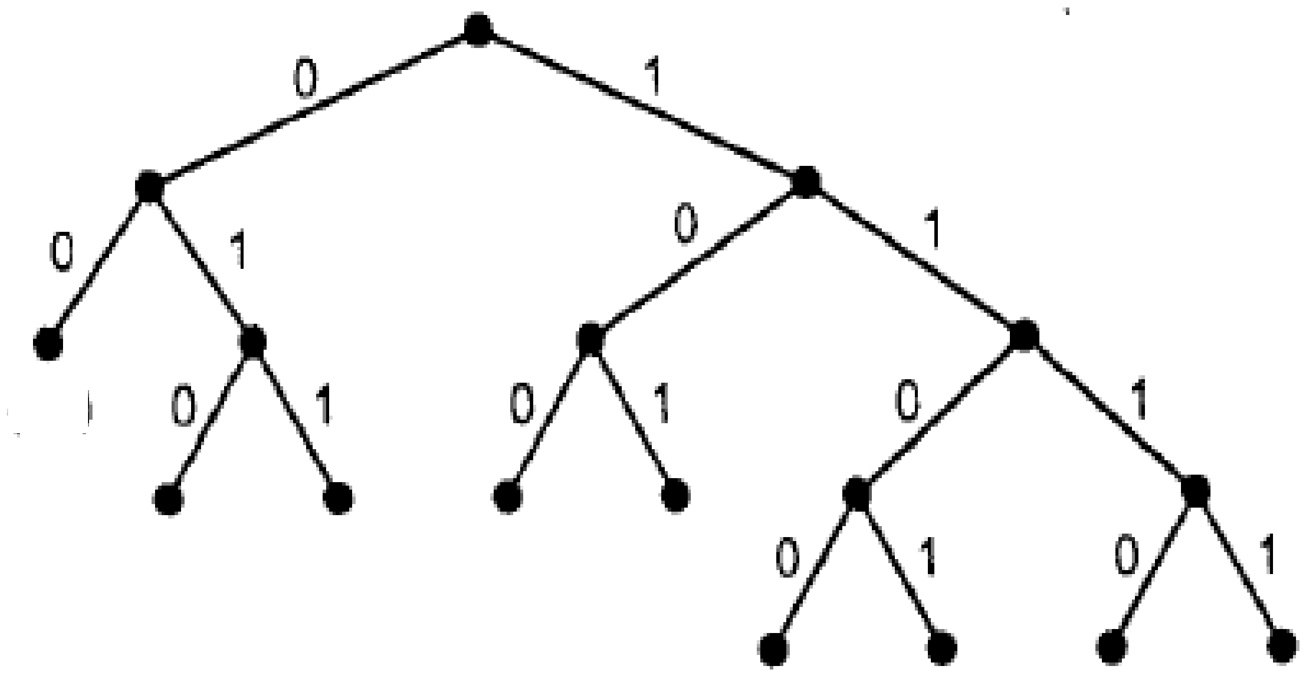
Using digits 0 and 1, we can represent a binary tree by binary prefix code and conversely from a binary prefix code, we can construct a binary tree. Consider a full binary tree in which from each vertex including the root there are two branches. Let us agree to assign 0 to the branch going to the left and assign 1 to the branch going to the right. In this way we go on assigning numbers to all edges and reach the terminal nodes. Terminal nodes then can be represented by sequences of 0, 1 as shown in the following diagram.

Example : Write binary prefix code for a given binary tree

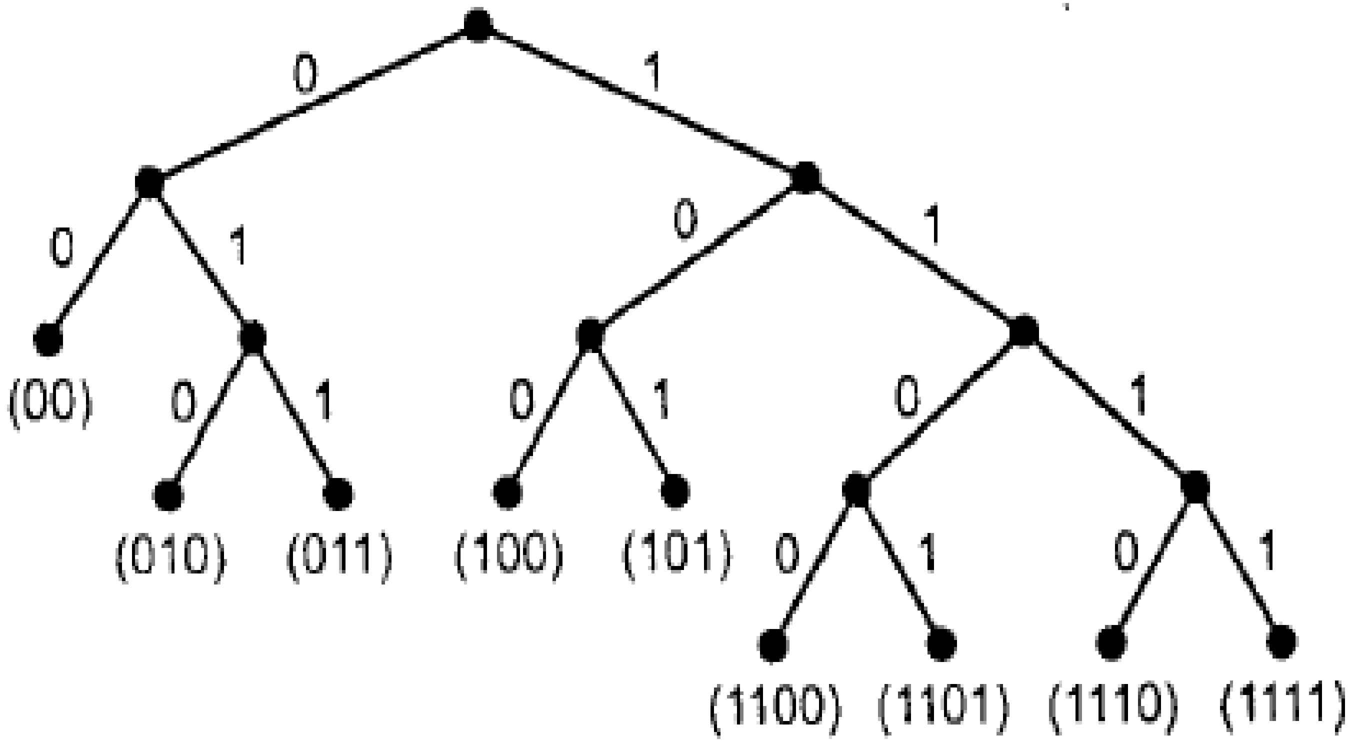
Example : Write binary prefix code for given binary tree



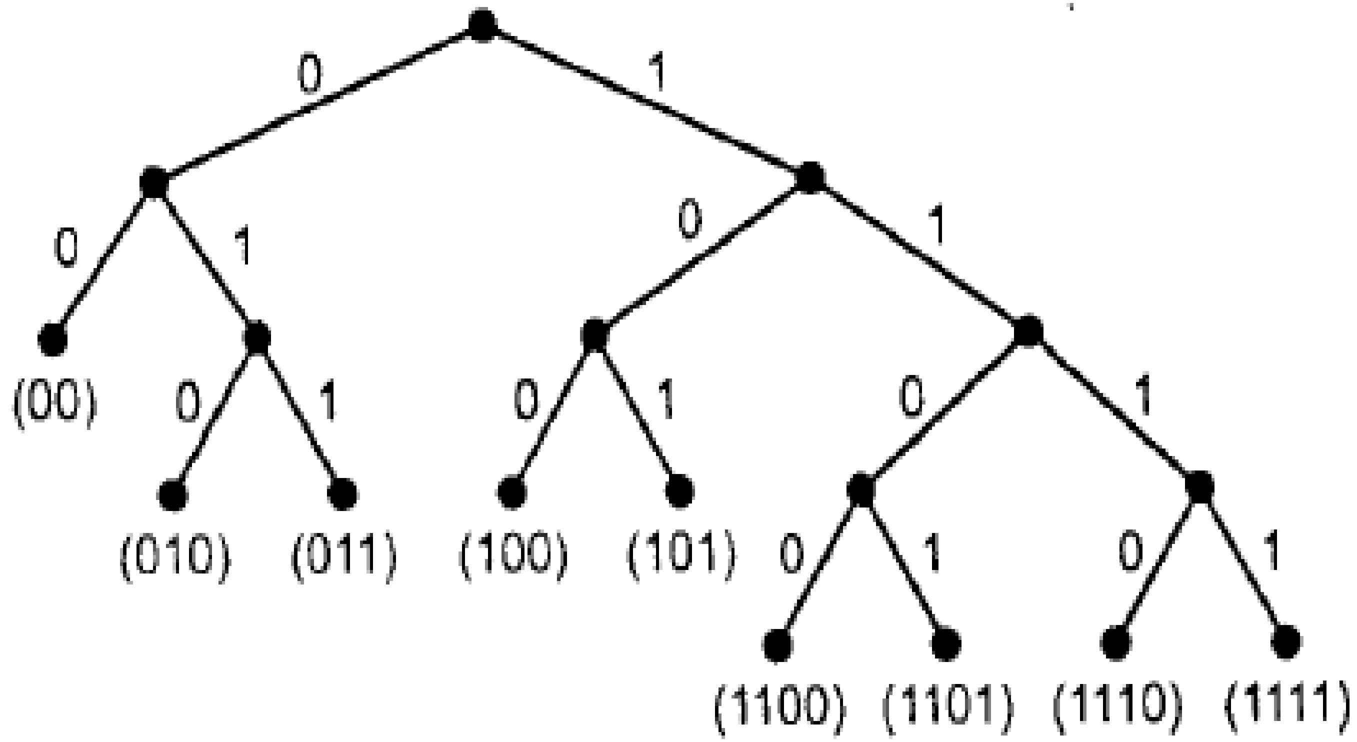
Example : Write binary prefix code for given binary tree



Example : Write binary prefix code for given binary tree



Example: Write binary prefix code for a given binary tree

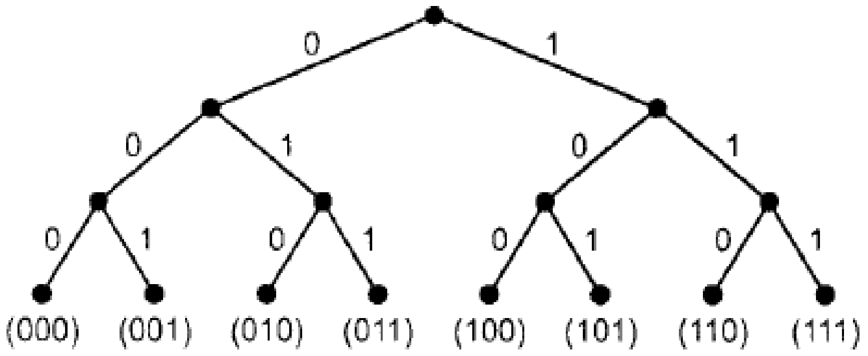


The prefix code of the above tree is

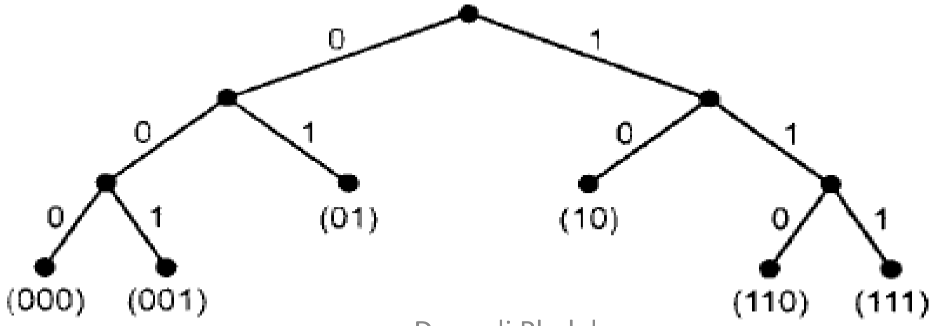
{ 00, 010, 011, 100, 101, 1100, 1101, 1110, 1111 }.

Example 1 : Draw a binary tree for binary prefix code {000,001,01,10,110,111}

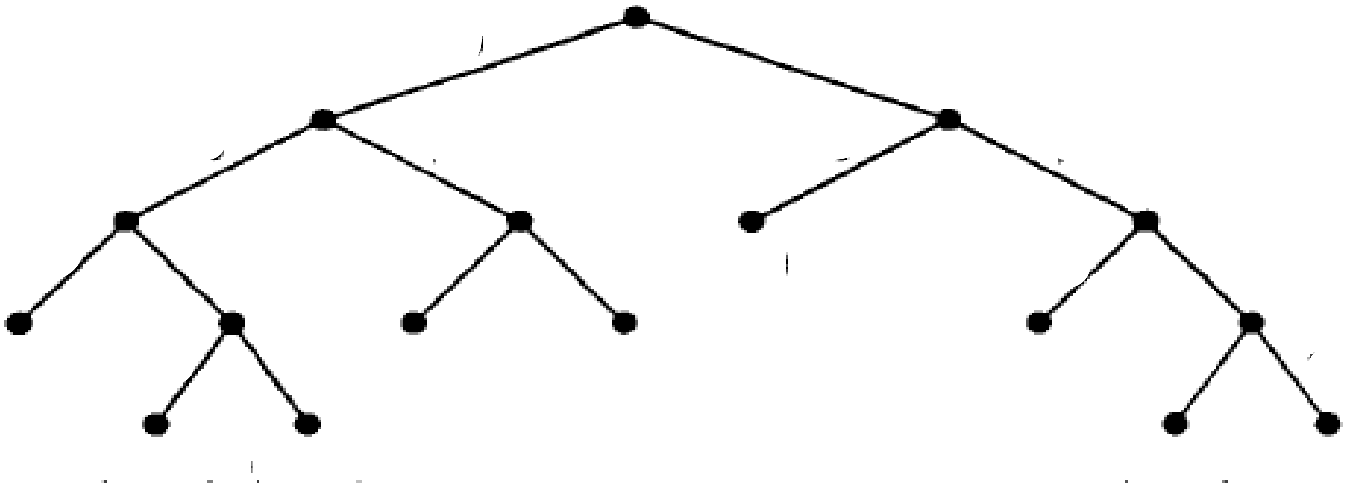
Since the longest sequence is of length three we first construct a full binary tree of level three as shown below.



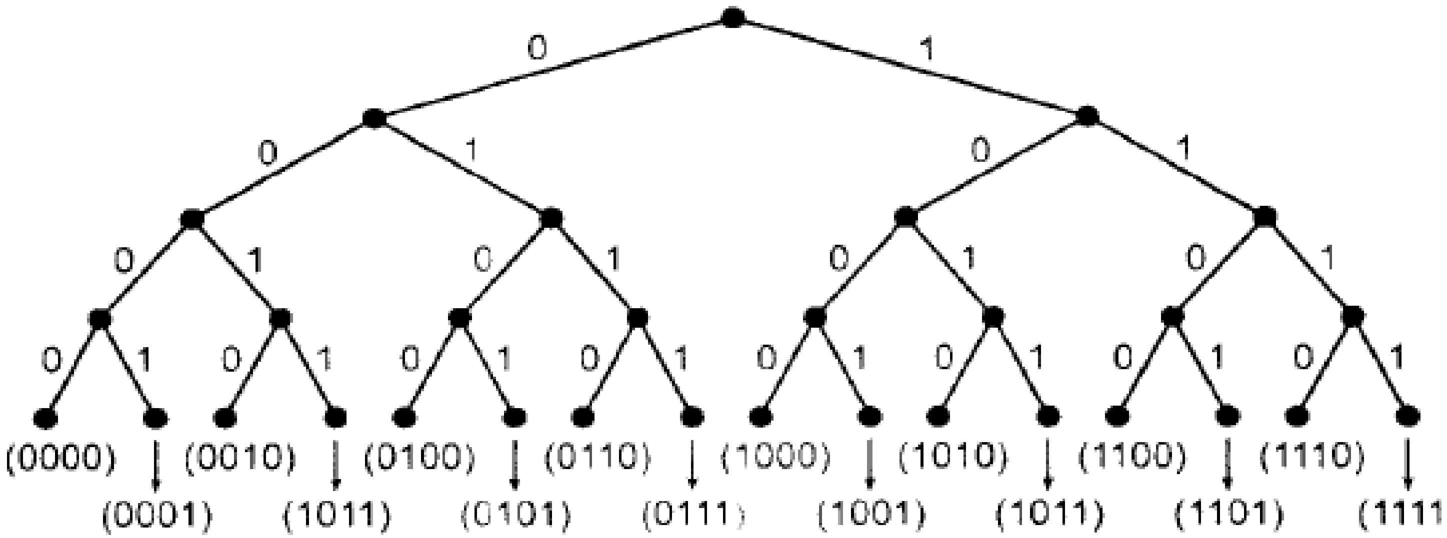
Now, we delete those vertices whose sequences are not included in the given prefix code. We also delete the incident edges. Thus, we get the following tree.

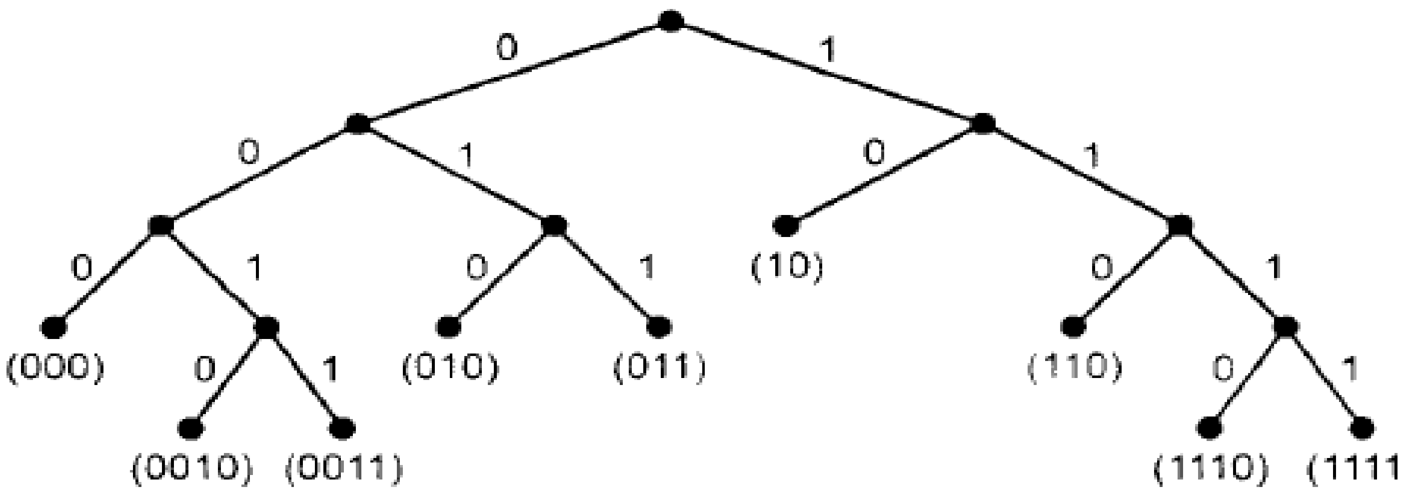


Example: binary prefix code for a binary tree is {10,000,010,011,110,0010,0011,1110,1111} Then draw Tree



Example 2 : Draw a binary tree for binary prefix code {10,000,010,011,110,0010,0011,1110,1111}





Optimal Binary Tree

Weight of a Tree

If terminal nodes of binary tree are given some weights, Say $w_1, w_2, w_3, \ldots\ldots w_k$ whereas they are located at level $l_1, l_2, l_3, \ldots\ldots l_k$ respectively. Then Weight of the Tree is given by, $W = \sum_{i=1}^{k} w_i l_i$

Huffman Procedure to find optimal binary tree for a given set of weights

Example: Obtain a optimal binary tree for weights 4, 3, 2, 11, 7. Also find Weight of Optimal Binary Tree.

Example: Obtain a optimal binary tree for weights 4, 3, 2, 11, 7. Also find Weight of Optimal Binary Tree.

ANS: Step 1 – Arrange weights in ascending order 2, 3, 4, 7, 11

Step 2 – Form subtree for two smallest Weights i.e. 2, 3 and 2+3 = 5

Step 3 – Now Consider 5, 4, 7, 11

Repeat Step 1 – 3.

Ascending - 4, 5, 7, 11

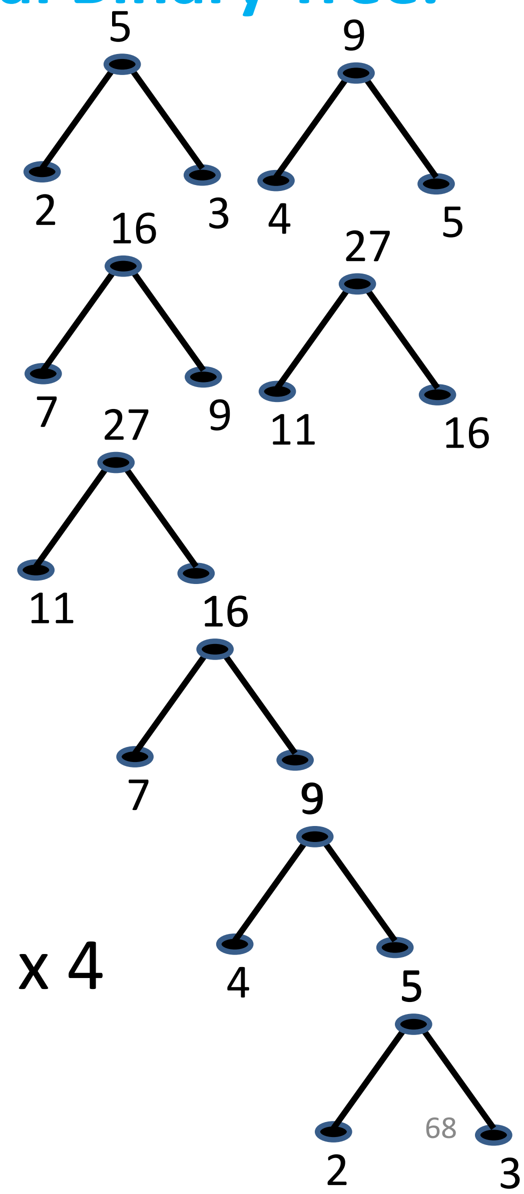
Subtree for 4, 5 and 4+5 = 9

Consider now 9, 7, 11 and so on.

Finally connect all subtree properly.

Weight = 11 x 1 + 7 x 2 + 4 x 3 + 2 x 4 + 3 x 4

Of OBT = 57



Coding and Decoding Using Binary Tree

Example : Using the code given by the following table encode the words (i) earn, (ii) year, (iii) treaty.

Character	:	*o*	*a*	*e*	*n*	*r*	*t*	*y*
Code	:	1100	1101	01	1110	10	0	1111

Sol. : To encode a word we just go on writing the codes of the characters in the word serially.

(i) 'earn' is coded as code of e then code of a, then code of r and lastly code n.

01 1101 10 1110

(ii) 'year' is coded as y + e + a + r

i.e. 1111 01 1101 10

(iii) treaty is coded as t + r + e + a + t + y

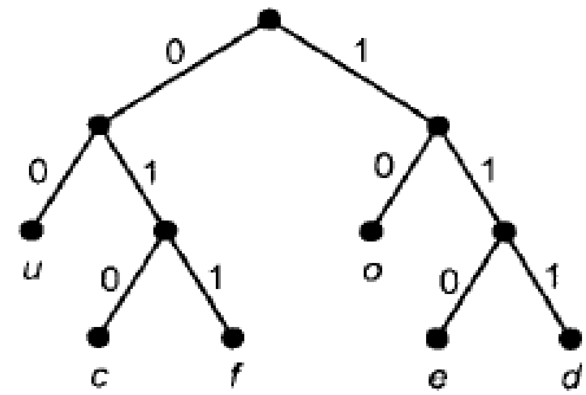
i.e. 0 1 0 0 1 1 10 1 0 1111

Using the Haffman tree decode the message 0101001101111 0110

From the tree, we first prepare the table of the Huffman code of the characters.

Character	:	*u*	*c*	*f*	*o*	*e*	*d*
Code	:	00	010	011	10	110	111

since the first digit is o, the first character is u, c or f.



Second digit is 1. Hence, c or f.

Third digit is 0, Hence c.

Fourth digit is 1. Hence, o, e or d.

Fifth digit is 0. Hence, o.

Sixth digit is 0. Hence, u, c or f.

Seventh digit is 1. Hence, c or f.

Eighth digit is 1. Hence, f.

Ninth digit is 0. Hence, u, c or f.

Tenth digit is 1. Hence, c or f.

Eleventh digit is 1. Hence, f.

Twelfth digit is 1. Hence, o, e or d.

Thirteenth digit is 1. Hence, e or d.

Fourteenth digit is 0. Hence, e.

The last sequence is 110 again. Hence, e.

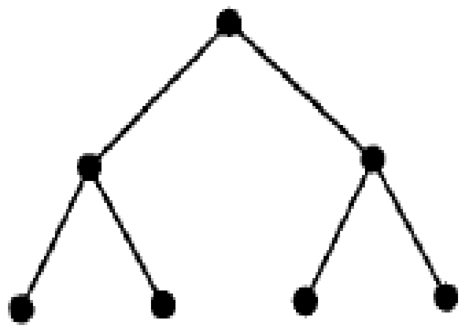
Hence, the message is 'coffee'.

Example :

Using the given Huffman tree decode the message

(i)1110110010111000

(ii)10111001111



Example :

Using the given Huffman tree decode the message

(i)1110110010111000

(ii)10111001111

ANS