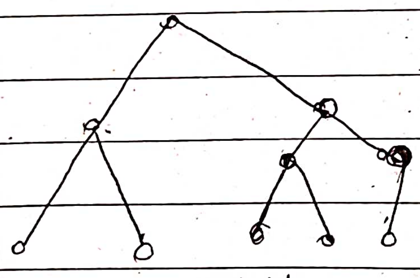
DT 306, 1007.

Connectedness in Undirected graph:

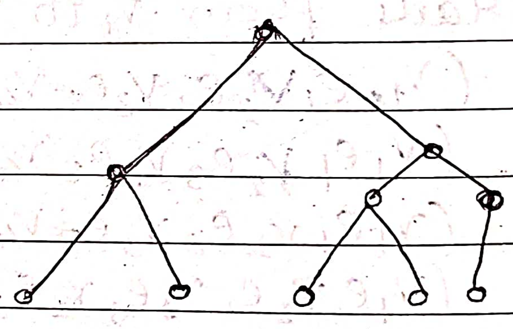
- An undirected graph $G = (V, E)$ is said to be connected if any pair of nodes are reachable from one another, i.e. if u and v are any two nodes then there exists a path between u & v. If a graph is not connected, then it is called disconnected graph.
- A disconnected graph is the union of two or more connected subgraphs each pair of which has no common node.
- These disjoint connected subgraphs are called connected components. A graph A is a connected ~~graph~~ component of a graph G iff (i) A is subgraph of G,
  (ii) A is connected.

(a)



Connected graph.

(b)



Connected graph.

disconnected graph.

(c)

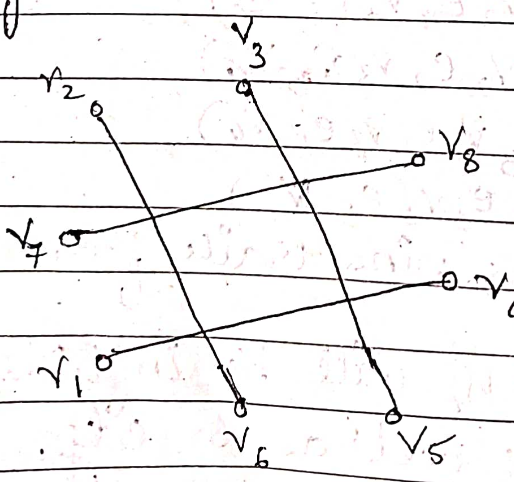
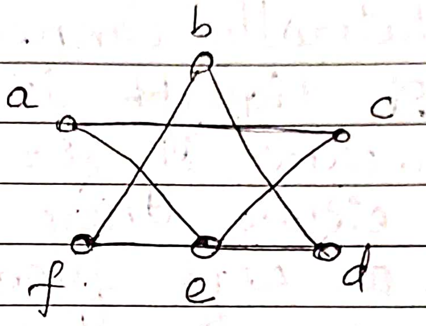
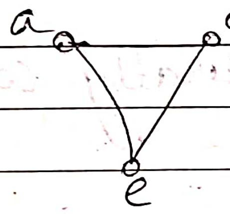
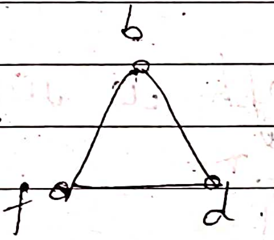


fig. Disconnected graph.

Consider the graph G.



Then 

& 

are the connected components of G.

Connectedness in diagraphs.

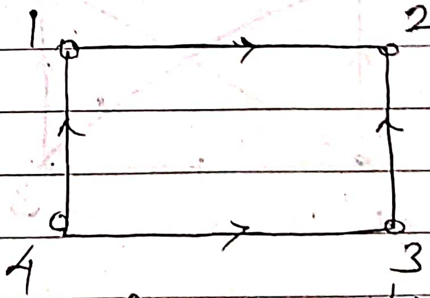
We ~~????~~ know that in a digraph, if there is an edge from vertex v to vertex u, then there need not be an edge from vertex u to vertex v.

IT S10,13,W11
CSE W10.

(1) Weakly connected:

A digraph is said to be weakly connected if it is connected as an undirected graph in which the direction of edge is neglected.

for eg. Consider the digraph



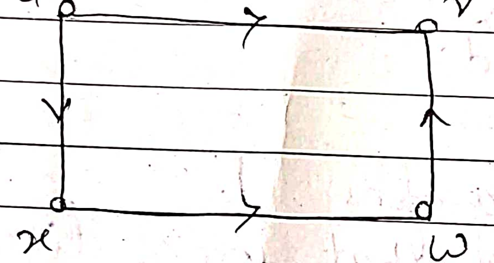
In this graph if we neglect the direction of edges, then it becomes an undirected graph & hence a weakly connected graph.

IT W09.

(ii) Unilaterally Connected :

A digraph is said to be unilaterally connected if for any pair of vertices u & v either there is an edge from u to v or an edge from v to u.

for eg. Consider the digraph is unilaterally connected

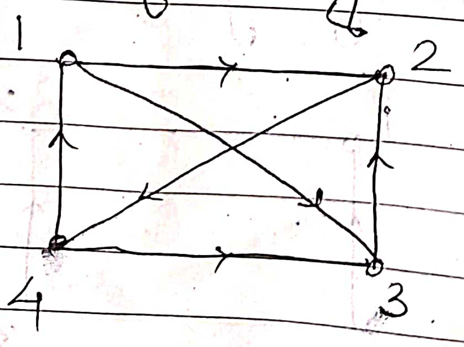


IT W09
CSE S09, 11, W09

(iii) Strongly connected :

A digraph is said to be strongly connected if for any pair of vertices u & v, there is a path from u to v and a path from v to u.

for eg. The following digraph is strongly connected.



~~IT~~ S506, W07
CSE W07, 11, S09

Euler Path :

Let $G=(V, E)$ be a connected graph. A path in G is called an euler path if it includes every edge exactly once. Since the path includes every edge exactly once, it is called euler trail.

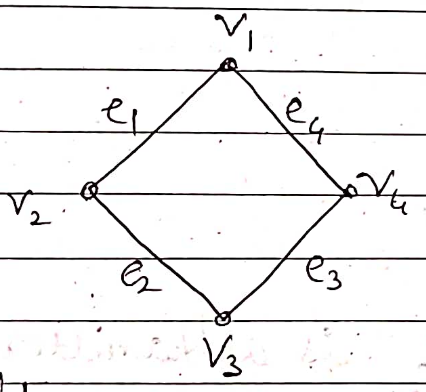
Euler circuit :

A euler path which is a circuit is called a euler circuit.

Euler graph :

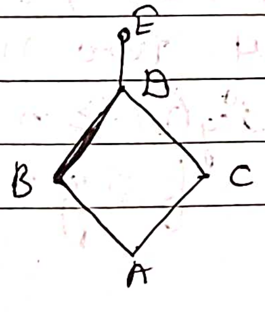
Let $G=(V, E)$ be a connected graph. Then G is called a euler graph if it contains a euler circuit.

for eg. : Consider the following graph.



Here $V_1 e_1 V_2 e_2 V_3 e_3 V_4 e_4 V_1$ is a euler path as well as euler circuit. Hence this a euler path.

ii) Now consider the given graph.



The eulerian path of the above graph is E-D-B-A-C.

CSE W11

Hamilton path :

Let $G = (V, E)$ be a connected graph. A path in G is called a Hamilton path if it contains every vertex exactly once.

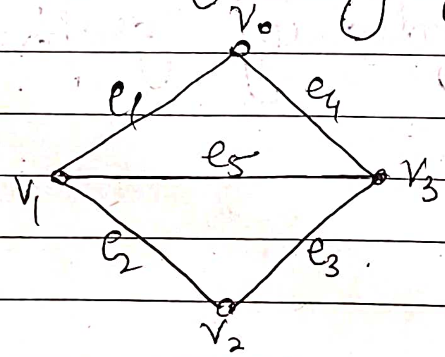
Hamilton circuit :

A Hamilton circuit is a circuit which contains every vertex exactly once except the first which appears twice.

Hamilton graph :

Let $G = (V, E)$ be a connected graph. Then G is called a Hamilton graph if it contains a Hamilton circuit.

for eg. Consider the following graph,



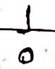
Hence $v_0 e_1 v_1 e_2 v_2 e_3 v_3 e_4 v_0$ is a Hamilton circuit
Hence this is a Hamilton graph.
Also $v_0 e_1 v_1 e_2 v_2 e_3 v_3$ is a Hamilton path.

Note: 1) Let G be a graph with n vertices, then G has Hamilton circuit if for any two non-adjacent vertices u & v, $\deg(u) + \deg(v) \geq n$.

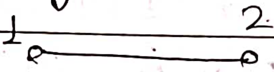
2) G has Hamiltonian circuit if $\deg(u) \geq n/2$ for each vertex u.

Tree:

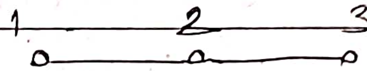
A directed tree (or simply a tree) is a connected acyclic graph i.e. a connected graph having no cycle. The edges of a tree are called branches. Following are some of the examples of trees.

i) 

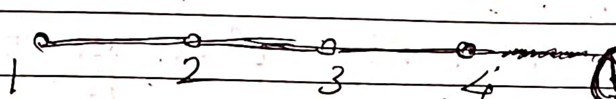
1 vertex

ii) 

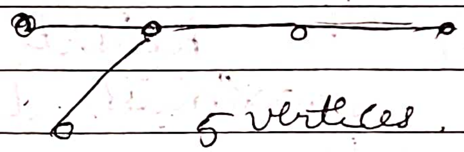
2 vertices

iii) 

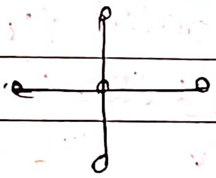
3 vertices.

(iv) 

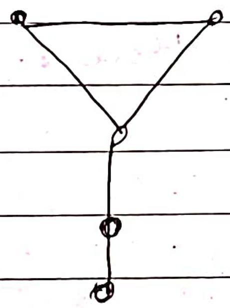
4 vertices.

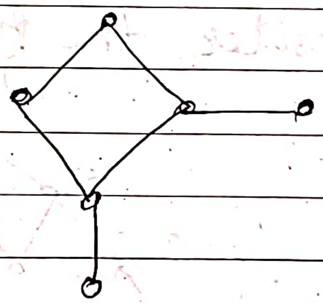
(v) 

5 vertices.

vi) 

The following graphs are not trees, because they contain a cycle.

(vii) 

(viii) 

Important results:

1) There is one and only one path between every pair of vertices in a tree T.
2) If in a graph G, there is one & only one path between every pair of vertices, then G is a tree.
3) A tree T with n vertices has $n-1$ edges.

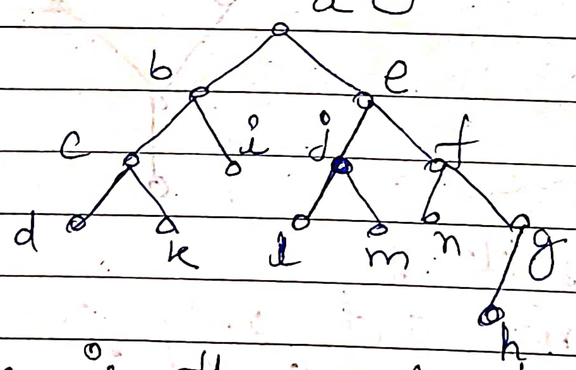
IT w09.

Rooted tree :

A rooted tree is a tree in which a particular vertex is called the root. Rooted trees in graph theory are drawn with their roots at the top.

1) The level of any vertex in a rooted tree is the length of the path from that vertex to the root. The level of the root is 0.
2) The height of a rooted tree is the maximum level ~~of~~ to any vertex of the tree.
3) Given any vertex v of a rooted tree, the children of v are the vertices that are immediate adjacent to v.
4) A vertex v having no children is called a leaf.

eg: Consider the following rooted tree :



Here a is the root, b and e are the children of a, n and g are the children of f

Vertex :	a	b	c	d	e	f	g	h	i	j	k	l	m	n
Level :	0	1	2	3	1	2	3	4	2	2	3	3	3	3

Here, height = maximum Level = 4

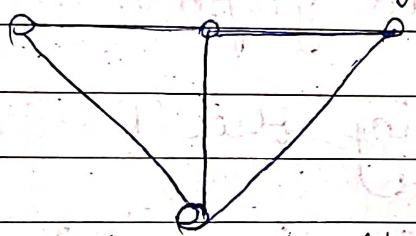
Here d, k, l, m, n, h, i are leaves

Spanning Tree :

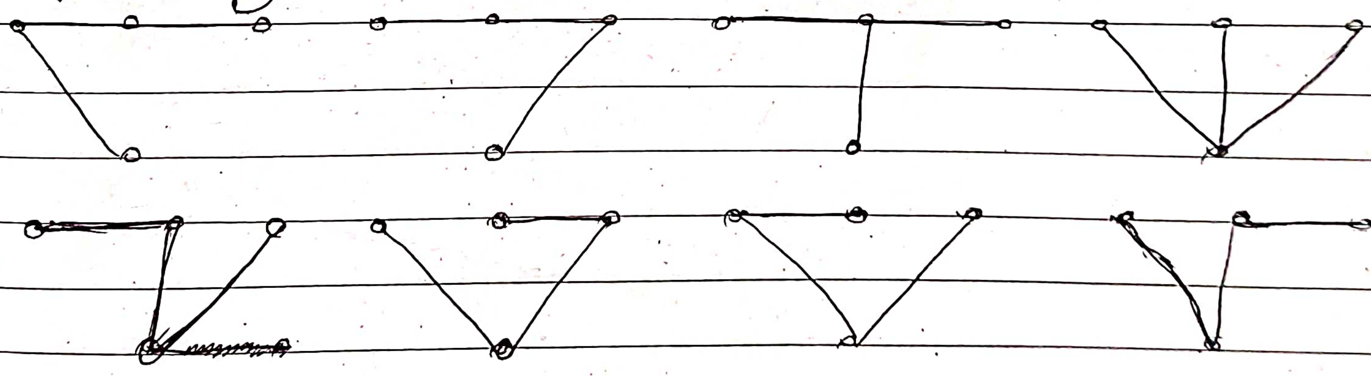
A subgraph T of a connected graph $G = (V, E)$ is called a spanning tree if T is a tree and T includes every vertex of G ie. $V(T) = V(G)$

Note: If $|V| = n$ and $|E| = m$, then the spanning tree of G must have n vertices and $n-1$ edges.

Eg. Find all the spanning trees of



Soln: Here the graph has 4 vertices and hence each spanning tree must have 4 vertices & $4-1=3$ edges. The required spanning trees are drawn below :



Note: A simple graph G has a spanning tree iff G is connected.

Minimal Spanning Tree:

Let G = (V, E) be a connected weighted graph. The weight of a spanning tree of G is the sum of the weights of the edges. A minimal spanning tree of G is a spanning tree of G with minimum weight.

5.17 Prim's Algorithm to find Minimal Spanning Tree :

Input : A connected weighted graph G.

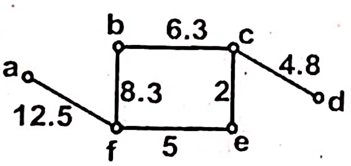
Output : A minimal spanning tree T.

Step (1) : Select an edge in G with minimum weight and select any vertex of this edge. Among all the edges incident with the selected vertex, select an edge with minimum weight. Include it in T.

Step (2) : At each stage, select an edge with minimum weight joining a vertex already included in T and a vertex which is yet to be included. (Make sure that a circuit is not formed). Include it in T.

Step (3) : Repeat this process until all the vertices of G are included in T.

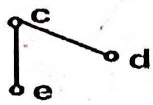
Example (26) : Find the minimal spanning tree of



using Prim's algorithm

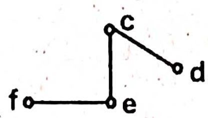
Solution: (1) Let us select vertex e as the edge (e, c) has minimum weight.

(2) Now there are two edges incident with c i.e. (c, d), (c, b) and w (c, d) = 4.8, w (c, b) = 6.3. Hence minimum weight is 4.8.



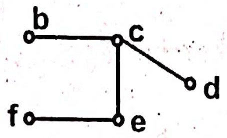
(3) Now there is one vertex incident with e i.e. (e,f) and one vertex incident with c i.e. (c, b).

w (e, f) = 5, w (c, b) = 6.3. Hence minimum weight is 5.

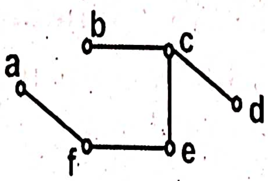


(4) Now there is one vertex incident with c i.e. (c, b) and two vertices incident with f i.e. (f, a) (f,b).

w(c,b) = 6.3, w (f, b) = 8.3, w (f, a) = 12.5 . Hence minimum weight is 6.3.



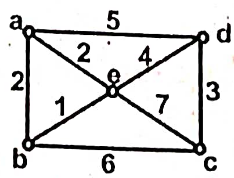
(5) Now only one vertex a incident with f is left.



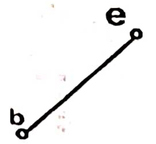
This is the required minimal spanning tree.

Its weight = 2 + 4.8 + 5 + 6.3 + 12.5 = 30.6

Example (27) : Using Prim's algorithm find the minimal spanning tree of (W-07)

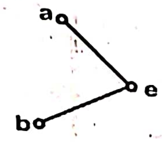


Solution : (1) Let us select vertex b as edge (b, e) has minimum weight.



(2) Now there are two edges incident with b i.e. (b,c) , (b, a) and three edges incident with e i.e. (e, a), (e,d),(e,c) and w (b, c) = 6, w (b, a) = 2, w (e,a) =2, w(e, d) = 4, w (e, c) = 7.

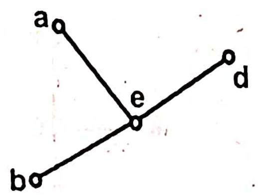
Hence minimim weight is 2.



(3) Now there is one edge incident b i.e. (b, c) and one edge incident with a i.e. (a, d) and two edges incident with e i.e. (e, c), (e, d).

w (b, c) = 6, w (a, d) = 5, w (e, c) =7, w (e, d) = 4.

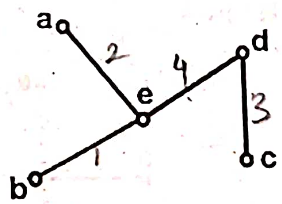
Hence minimum weight is 4.



(4) Now only one vertex c is left which is incident with vertices b, e, d

w (b, c) = 6, w(e, c) = 7, w (d, c) = 3.

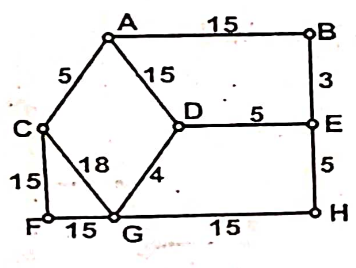
Hence minimum weight is 3.



(5) This is the required minimal spanning tree. (Its weight = 10).

Example (28) : Determine a railway network of minimal cost for the following cities.

(S-07,W-09)



Solution : (1) Let us select vertex B as edge (B, E) has minimum weight.

(2) Now there is one edge incident with B i.e. (B, A) and two edges incident with E i.e. (E, D), (E, H)

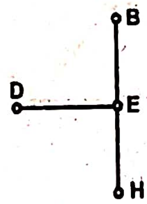
w (B,A) = 15, w (E, D) = 5, w (E, H) = 5

Hence minimum weight is 5.

(3) Now there is one edge incident with B i.e. (B, A) and one edge incident with E i.e. (E, D) and one edge incident with H i.e. (H, G).

w (B,A) = 15, w (E, D) = 5, w (H, G) = 15.

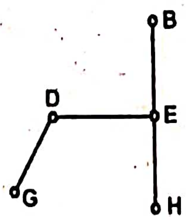
Hence minimum weight is 5.



(4) Now there is one edge incident with B i.e. (B, A) and one edge incident with D i.e. (D, G) and one edge incident with H i.e. (H, G) and w (D, G) = 4,

w (B, A) = 15, w (H, G) = 15.

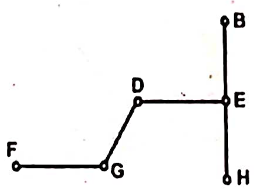
Hence minimum weight is 4.



(5) Now there is one edge incident with B i.e. (B,A) and two edges incident with G i.e. (G, F), (G, C)

w (B, A) = 15, w (G,F) = 15, w (G, C) = 18

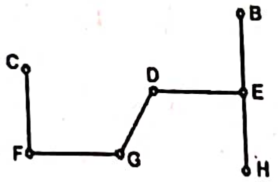
Hence minimum weight is 15



®

(6) Now there is one edge incident with F i.e. (F, C) and one edge incident with G i.e. (G, C) and one edge incident with B i.e. (B, A) and w (F, C) = 15, w (G, C) = 18, w (B, A) = 15.

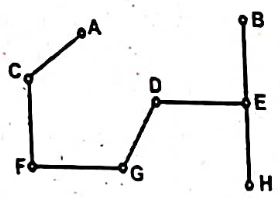
Hence minimum weight is 15.



(7) Now only one vertex A is left which is incident with B and C.

w (B,A) = 15, w (C, A) = 5.

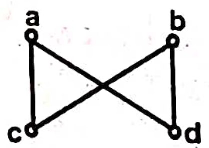
Hence minimum weight is 5.



This is the required minimal spanning tree.

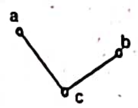
Minimal cost = 3 + 5 +5 + 4 + 15 + 15 + 5 = 52.

Example (29) : Using Prim's algorithm find the minimal spanning tree for the following graph using 'a' as the root. **(S-06;08,09)**

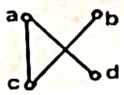


Solution : (1) Let us select vertex a. There are two edges incident with a i.e. (a, d), (a, c). Since the weights are not given, let us select edge (a, c).

(2) Now there is only one edge incident with a ie (a, d), and one edge incident with c ie (c,b). Since the weights are not given let us select edge (c, b).

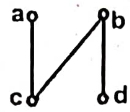


(3) Now only one vertex d is left which is incident with a and b ie (a, d), (b,d). Since the weights are not given let us select edge (a, d).



This is the required minimal spanning tree.

** In the above step (3), if we would have selected edge (b, d), then the minimal spanning tree would have been

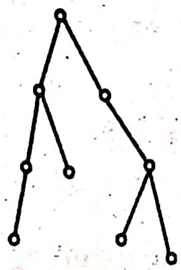


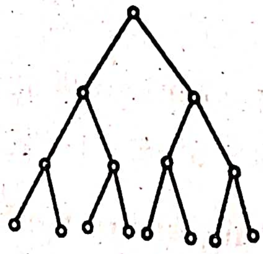
5.18 Binary Tree :

A binary tree is a rooted tree in which each vertex has atmost two children. That is a binary tree contains 0, 1 or 2 children. Each child is referred as either a left child or a right child (not both).

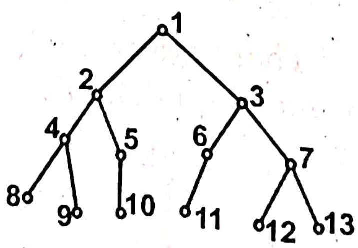
A **full binary tree** is a binary tree in which each vertex has exactly zero or two children.

Example (30) :

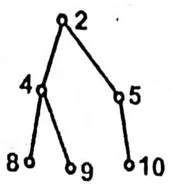

Binary Tree


Binary Tree

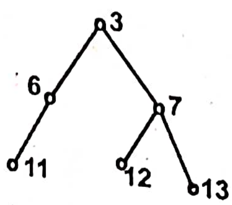
Example (31) : Find the left and right children of 2 in the following figure. What are the left and right subtrees of 1 ?



Solution : The left and right children of 2 are 4 and 5 respectively. The following trees represent the left and right subtrees of 1.



Left subtree of I.



Right subtree of I.