

VI . GRAPH THEORY

Graph:- A graph G has pair (V, E) where V is a non empty

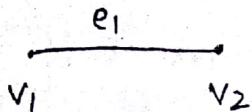
finite set whose elements are called vertices (nodes or points).

E is another set whose elements are called edges (lines).

The graph G with vertices V and edges E is written as

$$G = (V, E) \text{ or } G(V, E).$$

Ex:-



$$\text{Here } V = \{v_1, v_2\}, E = \{e_1\}$$

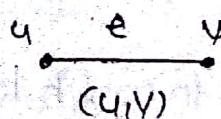
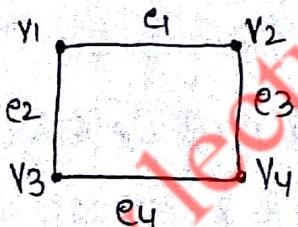
Note:- 1. If an edge $e \in E$ is associated with an ordered pair

(u, v) where $(u, v) \in V$.

2. e (edge) is connected to u and v and called end points of e .

3. Any two vertices connected by an edge in a graph is called adjacent vertices.

4. Any two edges e_1 and e_2 are incident with a common point (or vertex) then they are called adjacent edges.

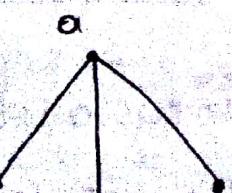


Here e has two adjacent vertices u and v .

In the above graph v_1 has two adjacent edges e_1 and e_2 and v_1 denoted as incident vertex.

$V = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (a, d)\}$ draw the graph G .

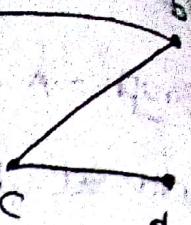
Sol:-



2. construct the vertices and edges from given graph.

Sol:- the given graph $G_1 = (V, E)$.

$$V = \{a, b, c, d\} \quad E = \{(a, b), (b, c), (c, d)\}$$



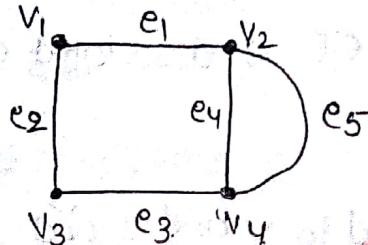
loop:- An edge of a graph G_1 that join a node to itself is called a loop (or) self loop defined as $e_1 = (v_1, v_1)$.



Ex-8-16

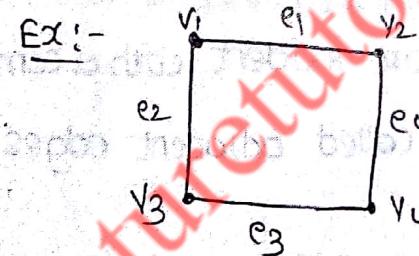
Multigraph:- If more than one line (edge) joining between two vertices are allowed in a graph then the graph is called multigraph.

Ex:-



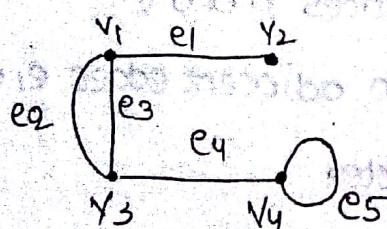
Simple graph:- A graph has neither loops nor multiple edges is called a simple graph.

Ex:-



Pseudo graph:- A graph in which loops and multiple edges are allowed is called pseudo graph.

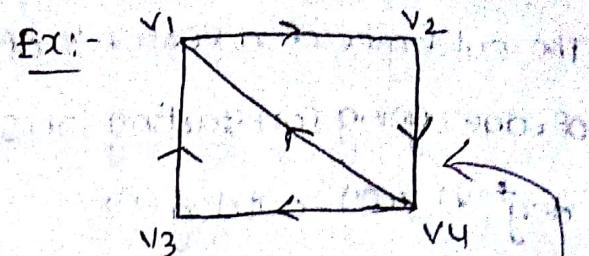
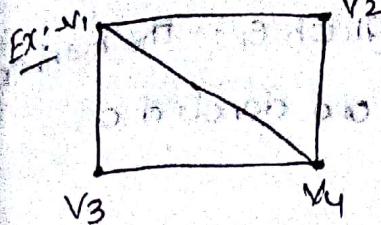
Ex:-



Directed and undirected graph:-

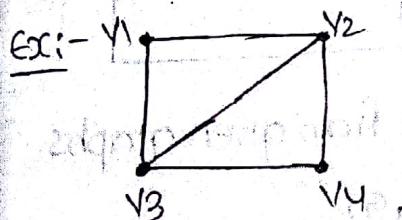
Undirected graph:- An undirected graph G_1 has a set of vertices V and a set of edges E such that each edge $e \in E$ is associated

with an unordered pair of vertices. ($e \in (v_i, v_j)$ and (v_j, v_i))



directed graph:- A directed graph G_1 has a set of vertices V and a set of edges E such that each edge $e \in E$ is associated with an ordered pair of vertices, means directions on each edge $e \in (v_i, v_j)$.

*degree of a vertex:- The degree of a vertex v of an undirected graph G_1 is the no. of edges incident with it. The degree of that vertex denoted as $\deg(v)$ or $d(v)$.



$$\deg(v_1) = 2 \quad \deg(v_3) = 3$$

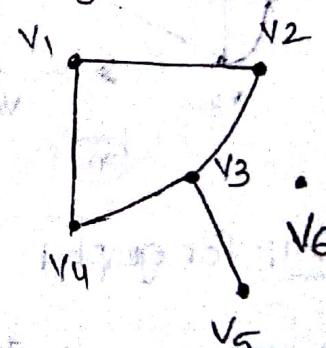
$$\deg(v_2) = 3 \quad \deg(v_4) = 2$$

1. construct degree of vertices from given diagram.

Sol:- $\deg(v_1) = 2, \deg(v_2) = 2$

$$\deg(v_3) = 3, \deg(v_4) = 2$$

$$\deg(v_5) = 1, \deg(v_6) = 0$$



Note:- 1. The vertex degree '0' is called Isolated vertex.

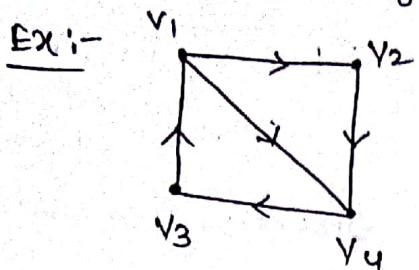
2. The vertex degree '1' is called pendant vertex.

3. In-degree and out-degree on directed graphs:-

The In-degree of a vertex v of a directed graph G_1 is the number of edges receiving (or) ending (or) coming at v and

denoted as $\deg^-(v)$ (or) $\text{Indeg}(v)$.

The outdegree of a vertex v of a directed graph G_1 is the number of edges going (or) starting (or) sending at v and denoted as $\deg^+(v)$ (or) $\text{outdeg}(v)$.



$$\deg^-(v_1) = 1$$

$$\deg^-(v_2) = 1$$

$$\deg^-(v_3) = 1$$

$$\deg^-(v_4) = 2$$

In-degree

$$\deg^+(v_1) = 2$$

$$\deg^+(v_2) = 1$$

$$\deg^+(v_3) = 1$$

$$\deg^+(v_4) = 1$$

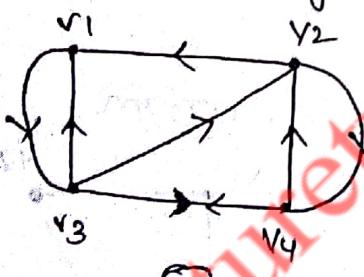
out-degree

Note:- If $G_1 = (V, E)$ is a directed graph with edge e then

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = E$$

$$\sum_{i=1}^n \deg(v_i) = 2E$$

1. Construct In-degrees and out-degrees from given graphs.



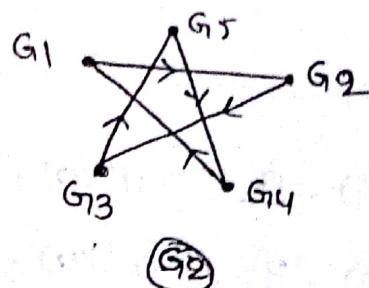
Sol:- for graph 1

$$\deg^-(v_1) = 2 \quad \deg^+(v_1) = 1$$

$$\deg^-(v_2) = 2 \quad \deg^+(v_2) = 2$$

$$\deg^-(v_3) = 2 \quad \deg^+(v_3) = 2$$

$$\deg^-(v_4) = 1 \quad \deg^+(v_4) = 2$$



for graph 2

$$\deg^-(G_1) = 1 \quad \deg^+(G_1) = 1$$

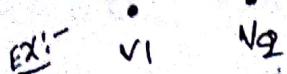
$$\deg^-(G_2) = 1 \quad \deg^+(G_2) = 1$$

$$\deg^-(G_3) = 1 \quad \deg^+(G_3) = 1$$

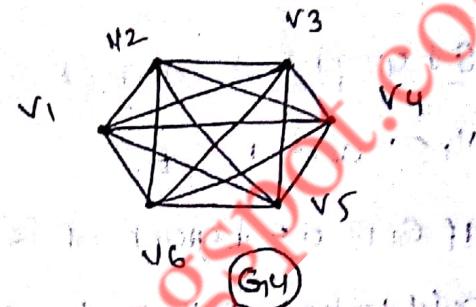
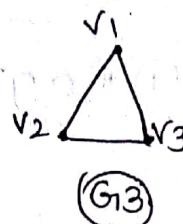
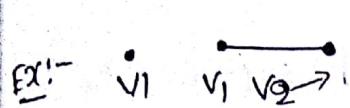
$$\deg^-(G_4) = 1 \quad \deg^+(G_4) = 1$$

$$\deg^-(G_5) = 1 \quad \deg^+(G_5) = 1$$

null graph:- A graph G_1 in which node (or) vertex is isolated. Node is called a null graph. The vertex v has '0' edges.



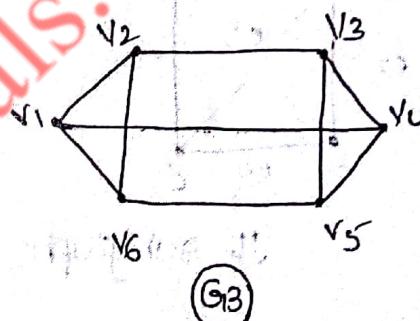
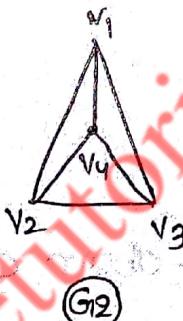
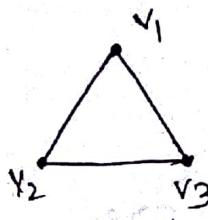
complete graph:- A simple graph G_1 is said to be complete graph if every vertex in G_1 is connected with every other vertex. i.e exactly one edge between pair of distinct vertices.



Regular graph:-

A graph G_1 has all vertices of degree is equal, is called a regular graph.

Ex:-

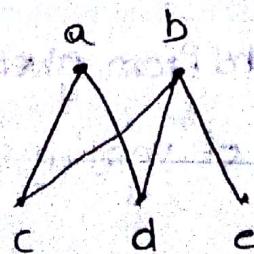
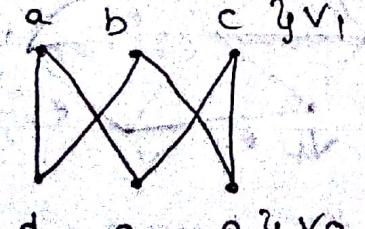


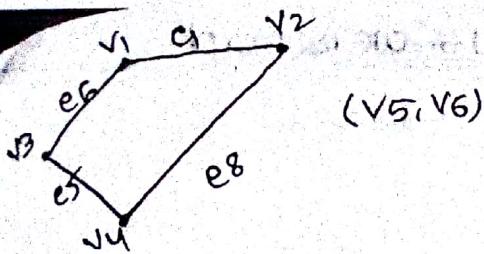
Q.9 :-

*Bipartite graph:- A graph $G = (V, E)$ is said to be Bipartite graph if the vertex V can be divided into 2 disjoint subsets

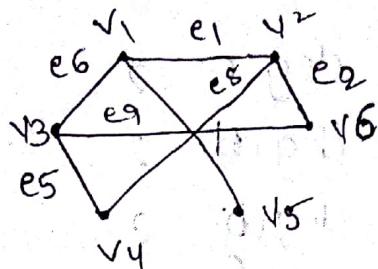
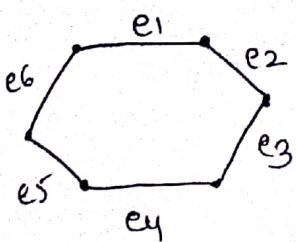
V_1 and V_2 such that every edge e connects from V_1 to V_2 . No edge is connects either two vertices in V_1 (or) V_2 of G .

Ex:-





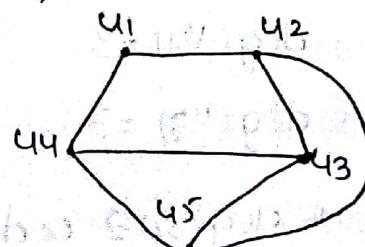
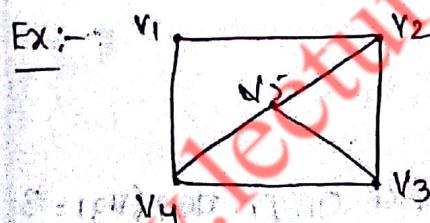
elimination of edges:-



Isomorphism:- Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be Isomorphic if there exists a Bijective function $\varphi : V_1 \rightarrow V_2$ such that $(u_1, v_1) \in E_1$ (or) adjacent vertices in G_1 . $(u_2, v_2) \in E_2$ are adjacent vertices in G_2 .

degree of vertex in G_1 are equivalent to degree of vertex in G_2 . If the adjacent vertices degrees are equal in G_1 and G_2 such that G_1 is isomorphic to G_2 , then we write as $G_1 \cong G_2$.

(degrees of vertices are same).



$$\deg(v_1) = 2 \Leftrightarrow \deg(u_1) = 2$$

$$\deg(v_2) = 3 \Leftrightarrow \deg(u_2) = 3$$

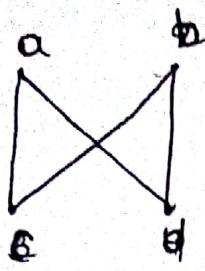
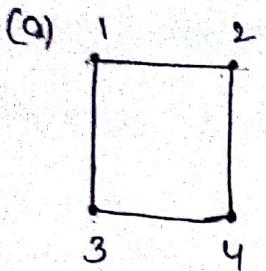
$$\deg(v_3) = 3 \Leftrightarrow \deg(u_3) = 3$$

$$\deg(v_4) = 3 \Leftrightarrow \deg(u_4) = 3$$

$$\deg(v_5) = 3 \Leftrightarrow \deg(u_5) = 3$$

\therefore The given two graphs are in Isomorphism.

1. Show that the following graphs G_1 and G_2 are isomorphic.



Sol:- $\deg(1) = 2 \Leftrightarrow \deg(a) = 2$

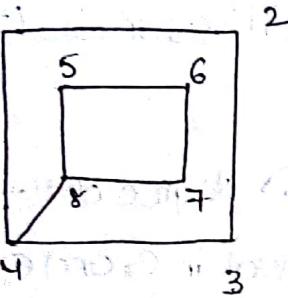
$\deg(2) = 2 \Leftrightarrow \deg(d) = 2$

~~$\deg(3) = 2 \Leftrightarrow \deg(b) = 2 \quad \deg(c) = 2$~~

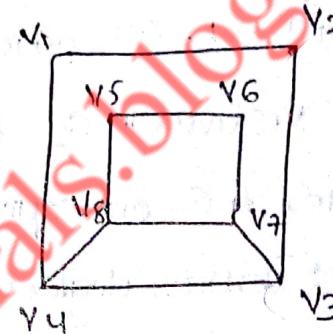
~~$\deg(4) = 2 \Leftrightarrow \deg(e) = 2 \quad \deg(f) = 2$~~

\therefore the given two graphs are isomorphic.

(b)



2



Isomorphic

$\deg(1) = 2 \Leftrightarrow \deg(V1) = 2$

$\deg(2) = 2 \Leftrightarrow \deg(V2) = 2$

$\deg(4) = 3 \Leftrightarrow \deg(V4) = 3$

$\deg(3) = 2 \Leftrightarrow \deg(V3) = 3$

In the first graph $\deg(3) = 2$ and in second graph $\deg(V3) = 3$

\therefore These two graphs are not isomorphic.

3) Matrix representation of a graph:- matrix representation of a graph has 2 types. 1. Adjacency matrix. 2. Incidence

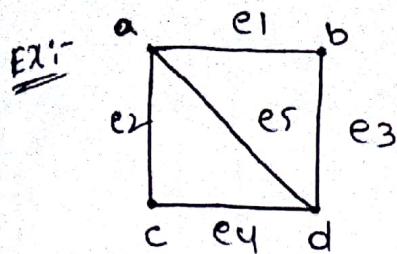
1. Adjacency matrix:-

Let $G = (V, E)$ be a simple graph with n vertices ordered from V_1 to V_n then the adjacency matrix $A_m = [a_{ij}]_{n \times n}$ of G is an $n \times n$

symmetric matrix defined by

$$a_{ij} = \begin{cases} 1 & \text{if } (V_i, V_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

$a_{ij} = \begin{cases} 1 & \text{When } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$



$$A_m = \begin{matrix} & a & b & c & d \end{matrix}$$

$$\begin{matrix} a & \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix} \\ b & \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \\ c & \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \\ d & \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

8.9-10

Q. Incidence matrix :-

Let G be a graph with n vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$

and $E = \{e_1, e_2, \dots, e_m\}$ define $n \times m$ matrix $I_m = [a_{ij}]_{n \times m}$

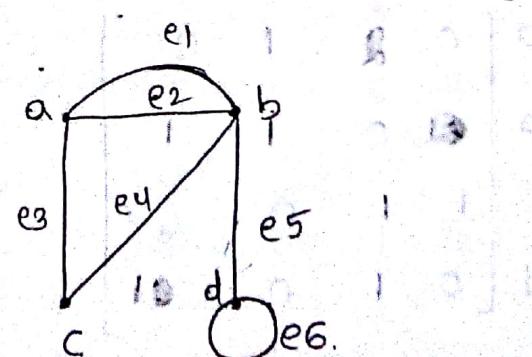
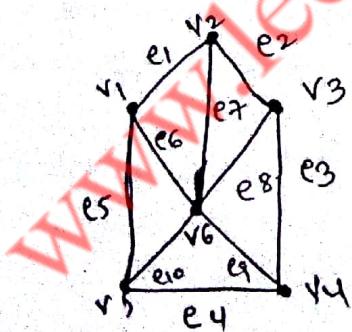
Where $a_{ij} = \begin{cases} 1 & \text{When } v_i \text{ is incident with } e_j \\ 0 & \text{otherwise} \end{cases}$

Ex:-

$$A_m = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix}$$

$$\begin{matrix} a & \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \end{bmatrix} \\ b & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} \\ c & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \end{bmatrix} \\ d & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

1. find the adjacency and incidence matrix from given graphs



Sol:- The adjacent matrix from first graph is $A_m =$

The incidence $\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix}$ is $I_m =$

$$A_m = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ v_1 & 0 & 1 & 0 & 0 & 1 & 1 \\ v_2 & 1 & 0 & 1 & 0 & 0 & 1 \\ v_3 & 0 & 1 & 0 & 1 & 0 & 1 \\ v_4 & 0 & 0 & 1 & 0 & 1 & 1 \\ v_5 & 1 & 0 & 0 & 1 & 0 & 1 \\ v_6 & 1 & 1 & 1 & 1 & 1 & 0 \end{matrix}$$

$$\bar{A}_m = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \\ v_1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_3 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ v_5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ v_6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{matrix}$$

The adjacent matrix to Second graph

$$A_m = \begin{matrix} & a & b & c & d \\ a & 0 & 2 & 1 & 0 \\ b & 2 & 0 & 1 & 1 \\ c & 1 & 1 & 0 & 0 \\ d & 0 & 1 & 0 & 0 \end{matrix}$$

$$\begin{matrix} a & 0 & 2 & 1 & 0 \\ b & 2 & 0 & 1 & 1 \\ c & 1 & 1 & 0 & 0 \\ d & 0 & 1 & 0 & 0 \end{matrix}$$

The Incidence matrix to second graph is

$$S_m = \begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \left[\begin{matrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{matrix} \right] \end{matrix}$$

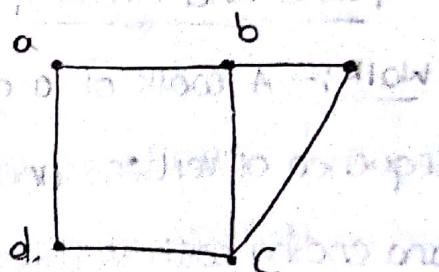
draw the graph represented by the adjacence matrix.

$$A_G = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \quad A_G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Sol:- The given adjacence matrix is

Let $A_G = \begin{matrix} & a & b & c & d & e \end{matrix}$

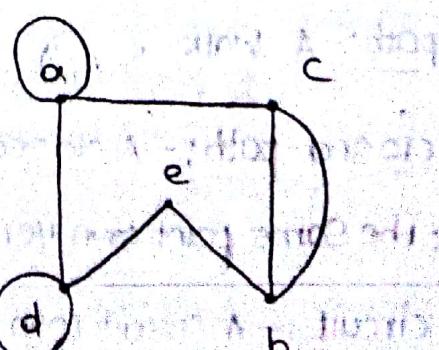
$$A_G = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$



The given adjacence matrix is

Let $A_G = \begin{matrix} & a & b & c & d & e \end{matrix}$.

$$A_G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



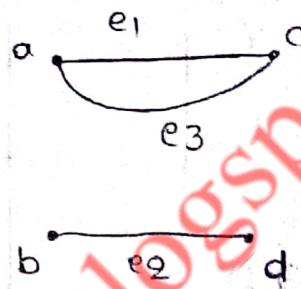
3. Draw the graph from given incidence matrix

$$IG = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Sol:- The given incidence matrix is

$$IG = e_1 \ e_2 \ e_3 \ e_4$$

$$\begin{array}{l} a \left(\begin{array}{cccc} 0 & 1 & 0 & 1 & 0 \end{array} \right) \\ b \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right) \\ c \left(\begin{array}{cccc} 1 & 0 & 1 & 0 \end{array} \right) \\ d \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right) \end{array}$$



Q-9-10

Paths and circuits:-

1. Walk:- A walk of a graph G is defined as an alternating sequence of vertices and edges. $v_0, e_1, v_1, \dots, e_n, v_n$. starting and ending with vertices such that each line e_i is incident with v_i . A walk joining v_0 and v_n is called v_0-v_n walk.

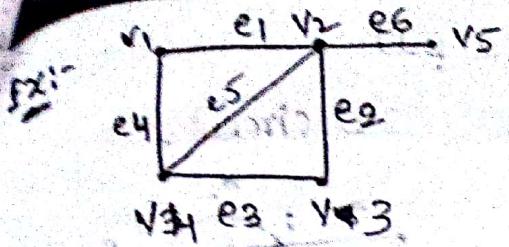
It contains only a single vertex such a walk is called trivial walk.

2. Trail:- A walk is called a trail if all its edges are distinct.

3. path:- A walk " " " path " " " vertices " "

4. closed path:- A closed path is a path that starts and ends at the same point (or) vertex.

5. circuit:- A circuit (or) cycle is defined as a closed path that does not contain repeated edges (distinct edges).



1. $v_1e_1v_2e_2v_3$

2. $v_1e_1v_2e_2v_3e_3v_4e_5v_2e_2v_3$

are walks.

3. $v_1e_1v_2e_5v_4e_3v_3e_2v_2e_6v_5$ is a trail.

4. $v_1e_4v_4e_3v_3e_2v_2e_6v_5$ is a path.

5. $v_1e_1v_2e_5v_4e_3v_3e_2v_2e_1v_1$ is a closed path

6. $v_1e_4v_4e_5v_2e_1v_1$

7. $v_1e_4v_4e_3v_3e_2v_2e_1v_1$ are circuits.

1. Determine of the following sequences are circuits & paths from below graph.

1. $v_1e_1v_2e_6v_4e_3v_3e_2v_2$

2. $v_1e_1v_2e_2v_3e_3v_4e_4v_5$

3. $v_1e_8v_4e_3v_3e_7v_1e_8v_4$

4. $v_5e_5v_1e_8v_4e_3e_2v_2e_6v_4e_4v_5$

5. $v_2e_2v_3e_3v_4e_4v_5e_5v_1e_1v_2$

Sol:- 1. vertex v_2 is repeated twice, so it is not a path.

= Starting vertex v_1 and ending vertex v_2 , so it is not circuit.

2. Here all vertices are distinct, so it is a path.

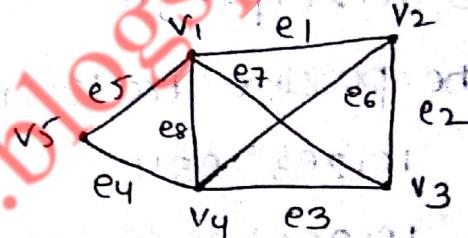
Starting vertex v_1 ending vertex v_5 , so it is not a circuit.

3. Here vertex v_1, v_4 are repeated, so it is not a path.

Starting vertex v_1 , ending vertex v_4 , so it is "Chall".

4. Here vertex v_5, v_4 are repeated, so it is not a path.

Starting vertex v_5 , ending vertex v_5 , so it is a circuit.

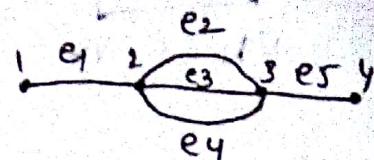


5. Here v_0 is repeated, so it is not a path.

Starting vertex v_2 , ending vertex v_2 , so it is a circuit.

Q. Let the graph G (i) How many paths

are there from 1 to 4. (ii) How many trails



12-9-10

Sol:- The possible paths are from 1 to 4 is

1. $1e_1 2e_3 3e_5 4$

2. $1e_1 2e_2 3e_5 4$

3. $1e_1 2e_4 3e_5 4$

The possible trails are from 1 to 4 is

1. $1e_1 2e_3 3e_5 4$

2. $1e_1 2e_2 3e_5 4$

3. $1e_1 2e_4 3e_5 4$

7. $1e_1 2e_4 3e_3 2e_2 3e_5 4$

9. $1e_1 2e_3 3e_2 2e_4 3e_5 4$.

4. $1e_1 2e_2 3e_3 2e_4 3e_5 4$

5. $1e_1 2e_2 3e_4 2e_3 3e_5 4$

6. $1e_1 2e_4 3e_2 2e_3 3e_5 4$

8. $1e_1 2e_3 3e_4 2e_2 3e_5 4$

* Eulerian graph (or) Euler graph (or) Eulerian circuit:-

1. A trail in G is called an Eulerian trail (distinct edges).

2. It contains all vertices atleast once of G .

3. A closed Eulerian trail (starting and ending Vertices Same) is

Called Eulerian graph (or) Euler circuit.

Euler path:-

1. A path in a Graph G is called an Euler path if it includes every edge exactly once (distinct edges).

2. visit all vertices at least once.

- Note:- 1. If G_1 is a graph in which the degree of every vertex is even then it is possible to construct Euler circuit.
2. The graph G_1 is a Euler path if atleast one degree of vertex is even.

3. If the given graph G_1 is not a Euler circuit and path, if and only if its vertices has odd degree.

1. Determine whether the graph is Euler path or circuit.

Sol:- From a given graph

Vertices $V = \{v_1, v_2, v_3, v_4\}$

Indegree to every vertex is

$$\deg(v_1) = 2, \deg(v_2) = 1, \deg(v_3) = 1, \deg(v_4) = 1$$

outdegree to every vertex is

$$\deg^+(v_1) = 1, \deg^+(v_2) = 1, \deg^+(v_3) = 2, \deg^+(v_4) = 1$$

Here vertex v_1 and v_3 has odd degree = 3

\therefore The Euler path is $v_3 - v_2 - v_1 - v_3 - v_4 - v_1$.

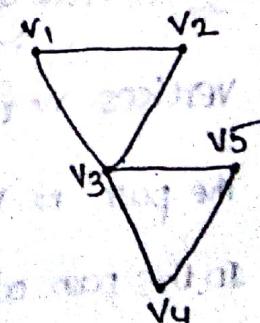
2. from the given graph check whether Euler circuit or path.

Sol:- From given graph

Vertices $V = \{v_1, v_2, v_3, v_4, v_5\}$

$$\deg(v_1) = 2, \deg(v_2) = 2, \deg(v_3) = 4$$

$$\deg(v_4) = 2, \deg(v_5) = 2$$

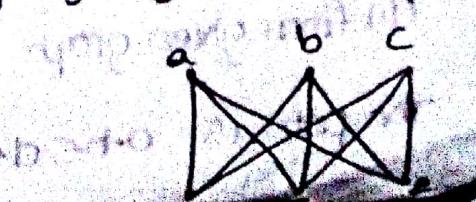


Here all the vertices have even degrees

\therefore The Euler circuit is $v_1 - v_2 - v_3 - v_4 - v_5 - v_3 - v_1$.

Sol:- From given graph

Vertices $V = \{a, b, c, d, e, f\}$



$\deg(a) = 3, \deg(b) = 3, \deg(c) = 3$

$\deg(d) = 3, \deg(e) = 3, \deg(f) = 3.$

Here all the vertices have odd degrees = 3.

The path is a-d-c-f-b-e-a-f

Here b-d, e-c edges are not covered. So it is not a path.

So it is not a circuit.

13-9-10

Hamiltonian graph: - A circuit in a graph G is called Hamiltonian circuit (or) graph.

2. If it contains each vertex in G exactly once except for the starting and ending vertex that appears twice.

Hamiltonian path: - A Hamiltonian path is a path that contains all vertices of G, where the endpoints (starting and ending vertices) may be distinct.

1. Determine which of the following graph is Hamiltonian circuit or path.

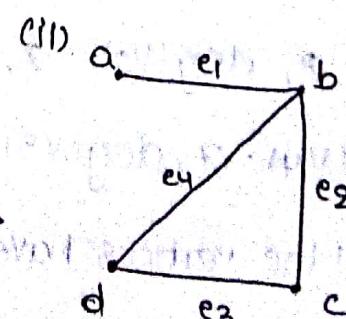
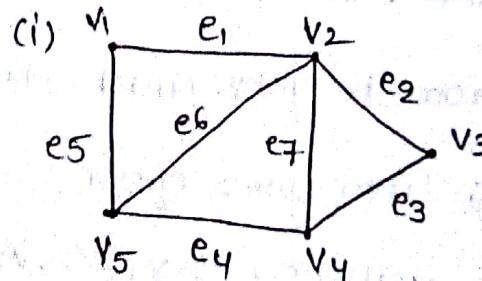
Sol: - (i) from given graph

Vertices $V = \{v_1, v_2, v_3, v_4, v_5\}$

The path is $v_1-v_2-v_3-v_4-v_5-v_1$.

In this path all vertices visited exactly once except starting & ending vertex.

\therefore It is Hamiltonian circuit.



(ii) from given graph Vertices $V = \{a, b, c, d\}$

The path is a-b-c-d.

$\{a, b, c, d\} \rightarrow V$ contains

In this path all vertices visited exactly once and starting and ending vertices are distinct.

\therefore It is a Hamiltonian path.

2. the given graph which is Hamiltonian circuit (or) Euler circuit

gol:- (i) case1:- from given graph

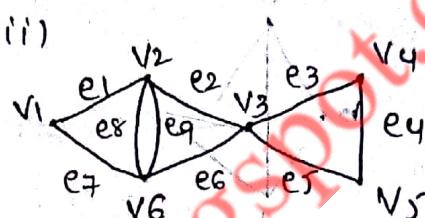
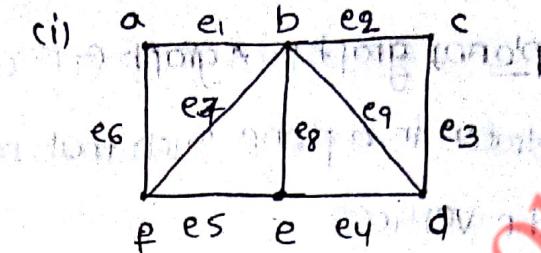
① $V = \{a, b, c, d, e, f\}$

The path is $a-b-c-d-e-f-a$

In this path all vertices visited

exactly once except starting and ending vertex.

\therefore It is a Hamiltonian circuit.



case2:- for given graph degrees to all vertices

$$\deg(a) = 2, \deg(b) = 5, \deg(c) = 2, \deg(d) = 3, \deg(e) = 3$$

$$\deg(f) = 3.$$

degree of vertices b, d, e, f has odd degree. It is not possible to construct Euler circuit.

(ii) Case1:- from given graph $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

The path is $v_1 - v_2 - \dots - v_6 - v_3 - v_4 - v_5$.

Here all vertices visited exactly once but starting and ending vertices are distinct.

\therefore It is not a Hamiltonian circuit.

case2:- from given graph degrees to all vertices

$$\deg(v_1) = 2, \deg(v_2) = 4, \deg(v_3) = 4, \deg(v_4) = 2$$

$$\deg(v_5) = 2, \deg(v_6) = 4$$

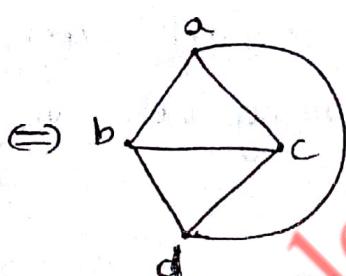
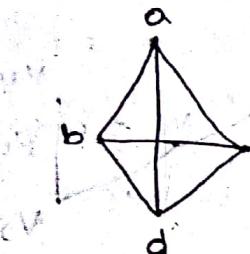
The path is $v_1 - v_2 - v_3 - v_4 - v_5 - v_3 - v_6 - v_2 - v_6 - v_1$.

Here all edges exactly once and visited all vertices. Starting and ending vertex is same.

\therefore It is Euler circuit

Planar graph:- A graph G_1 is called a planar graph if it can be drawn in a plane such that no two edges intersect except at the vertices.

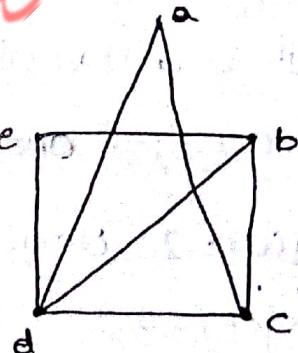
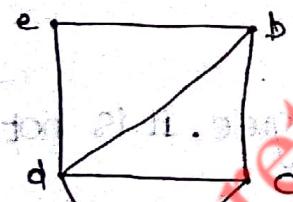
Ex:-



15-9-10

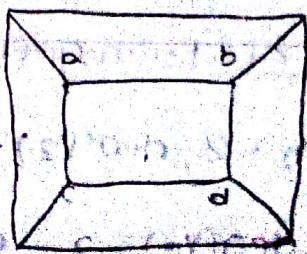
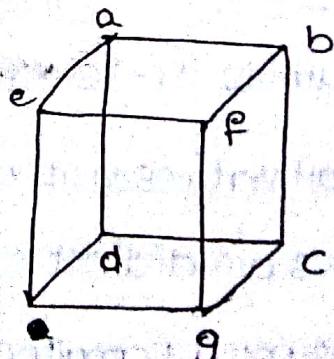
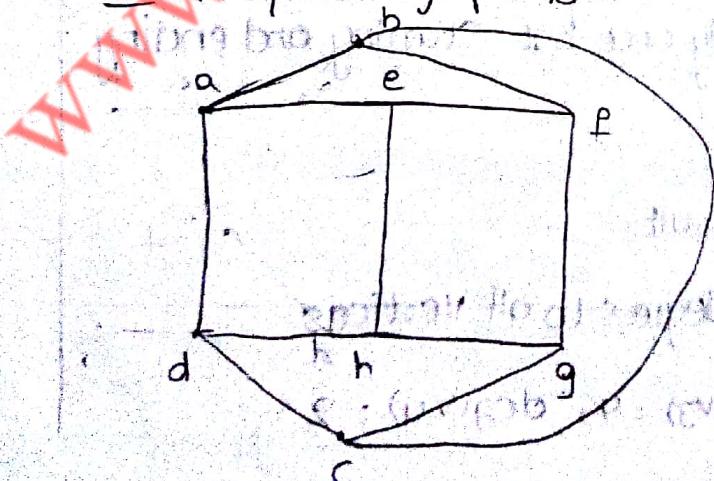
1. Determine the graph is it planar (or) not.

Sol:- The planar graph is



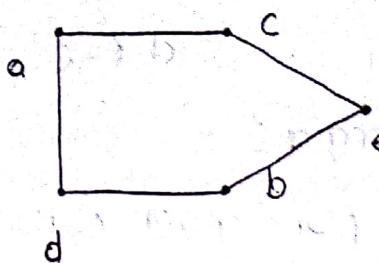
2. the given graph is planar (or) not.

Sol:- The planar graph is



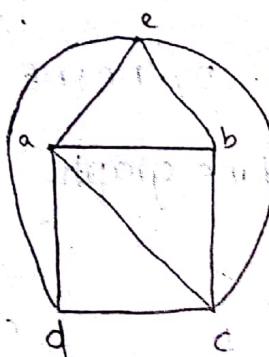
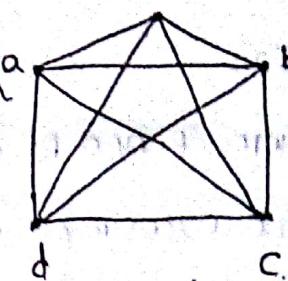
Q. The given two graphs are planar or not.

Sol:- The planar graph is



The planar graph is

It is a planar graph from given graph.



It is not possible to design the b to d edge. It is a nonplanar graph.
Every complete graph and above are equal to 5 vertices
that is not possible to design the planar graph.

∴ These are non planar graphs.

Graphs exterior and convexity:-

Characteristics:- $nv = ne + nf$

Where nv = number of vertices in a graph.

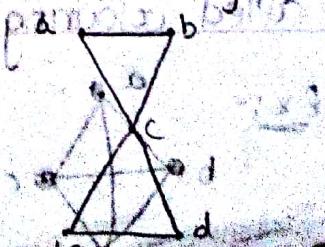
ne = " " edges

nf = " " faces (or) regions

nf :- It is the combination of interior and exterior regions in a plane graph.

(i) no. of vertices $nv = 5$

" " edges $ne = 6$



no. of faces (or) regions from given graph $n_f =$

$$R_1 = a-b-c-a, R_2 = c-d-e-c$$

R_1 = the region bounded by the cycle $a-b-c-a$.

R_2 = " " " " " " " " $c-d-e-c$.

Here R_1 and R_2 are interior regions.

The exterior region R_3 = the plane graph outside path

$$a-b-c-d-e-c-a.$$

$$\therefore n_f = 3.$$

The characteristics to given graph $n_v - n_e + n_f = 5 - 6 + 3 = 2$.

2. construct the characteristics to plane graph.

Sol:- no. of vertices $n_v = 4$

(d)

no. of edges $n_e = 4$

no. of faces (or) regions from given graph $n_f =$

$$R_1 = d-a-c-b-a.$$

Here R_1 is interior region

The exterior region R_3 = the plane outside path $a-c-b-a$

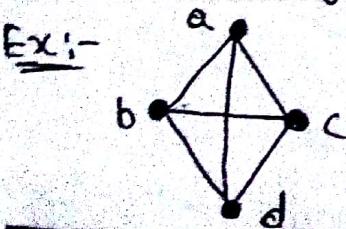
$$\therefore n_f = 2$$

The characteristics to given graph $n_v - n_e + n_f = 4 - 4 + 2 = 2$.

Graph colouring and covering:-

Colouring :- An assignment of colours to the Vertices of a graph and no two adjacent vertices get the same colour is called colouring of the graph (or) vertex colouring.

Ex:-



$f(a) = \text{Red}$

$f(b) = \text{Blue}$

$f(c) = \text{pink}$

$f(d) = \text{Yellow}$

chromatic number:- The chromatic number of a graph G_1 is the minimum number of colours needed to colour the vertices of the graph G_1 and denoted by $\chi(G_1)$.

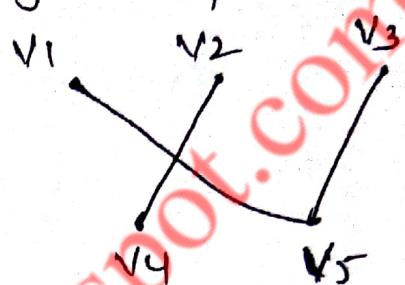
$$\therefore \chi(G_1) = 4$$

1. Determine the chromatic number of given Bipartite graph.

Sol:- $\therefore f(v_1) = R, f(v_2) = R, f(v_3) = R$

$$f(v_4) = B, f(v_5) = B$$

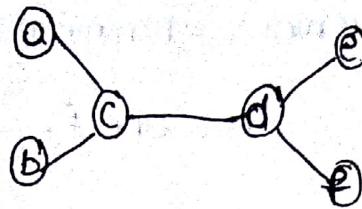
$$\chi(G_1) = 2$$



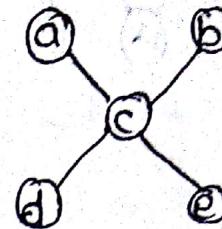
A tree is a simple graph G_1 such that there is a unique simple undirected path between each pair of vertices in G_1 .

Ex:-

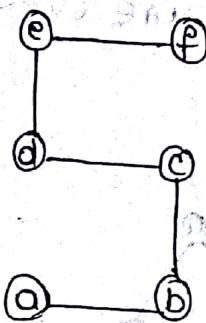
1.



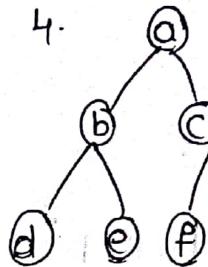
2.



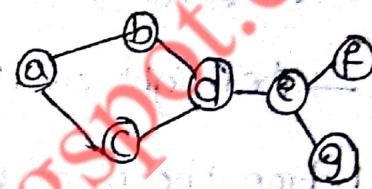
3.



4.



5.



Here 1, 2, 3, and 4 are trees. 5 is not a tree.

→ tree is denoted as ' T '.

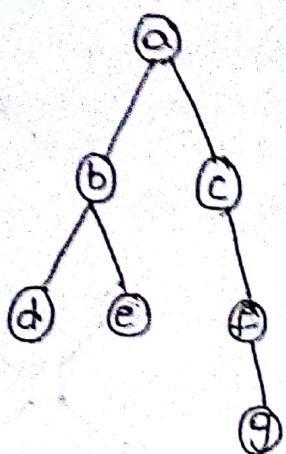
rooted tree:- A rooted tree is a tree in which a particular vertex is designated as the root (starting node or vertex).

→ If a vertex v of ' T ' is a child vertex, if that vertex is a end vertex (or) exit vertex. Such that v_0 as a root node. $(v_0, v_1, v_1, v_2, \dots, v_n)$ and v_n is a child vertex in every path.

→ except root and child vertices remaining all vertices are interval (or) middle vertices.

→ The level of a vertex v in a tree is the length of simple path from the root. The height of a rooted tree is the maximum level number.

Ex:-



In the above tree 'a' is root node
(or) vertex.

Here child nodes are d, e, and g.

Interval (or) terminal vertices are
b, c, and f.

\rightarrow a, b, c, d, e, f and g has 0, 1, 1, 2, 2, 2, 3 are levels respectively
to all vertices.

\rightarrow The height of tree is '3'.

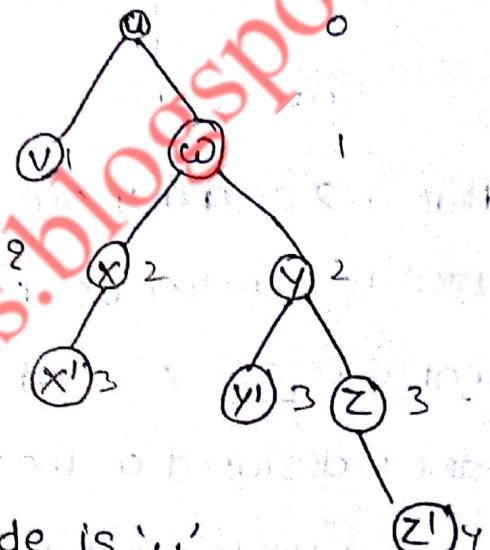
i. From the given rooted tree 'T'

(i) What is the root of T?

(ii) find the levels & interval vertices?

(iii) what are the levels of w and z?

(iv) find the childs of W and z?



Sol:- (i) from the given tree root node is 'u'.

(ii) The levels of given tree u, v, w, x, y, z, x', y', z'

0, 1, 1, 2, 2, 3, 3, 3, 4.

The interval vertices are u, v, w, x, y, z.

(iii) the level of w is 1 and level of z is 3.

(iv) the childs of W and z are x', y', z' and z'.

Spanning tree:- A tree 'T' is a Spanning tree of a graph G.

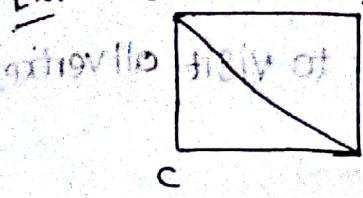
T is a subgraph of G that contains all of the vertices of G.

If G is a connected graph with n vertices and m edges,

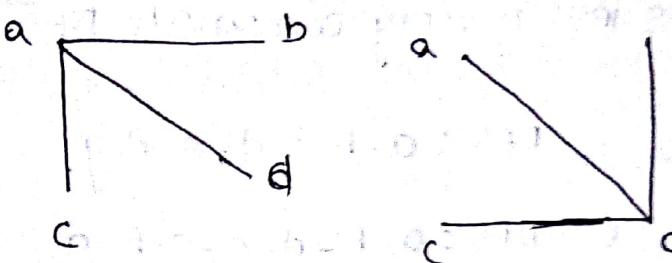
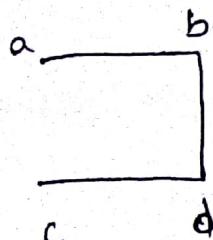
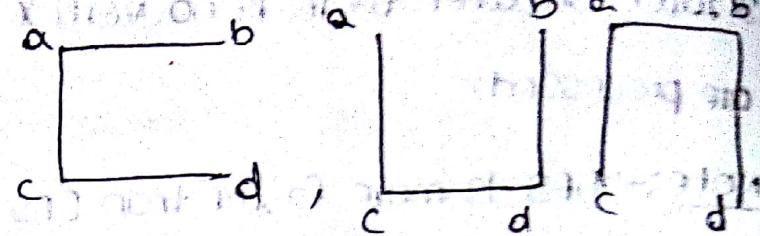
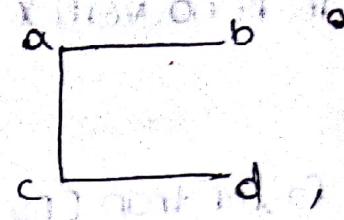
a spanning tree of G must have n-1 edges.

$$m=n-1$$

Ex-10 a

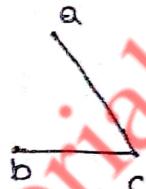


The spanning trees are



Q. Determine the spanning trees from given graph

Sol:- The spanning trees are



BFS and DFS in spanning trees :-

BFS (Breadth first search) :-

1. BFS starts traversal from root node and then explore the

Search in the level by level that is as possible from root node.

2. BFS can be done with the help of queue (first in first out).

3. The BFS works in a single level then visited vertices are removed from the queue until all vertices in tree.

DFS (Depth first search) :-

1. DFS starts from the traversal from the root node and explore

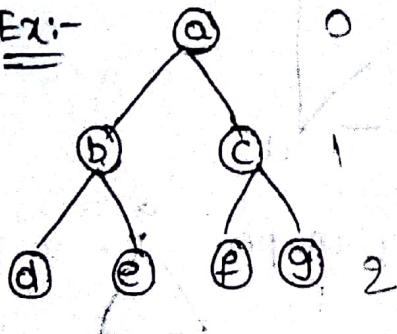
the search for as possible from the root node that is depth wise

2. DFS can be done with the help of stack (last in first out),
3. Later on, when there is no vertex further to visit all vertices are processed.

Note :- 1. DFS is more faster than BFS.

2. DFS requires less memory compare to BFS.

Ex:-



BFS : a - b - c - d - e - f - g,

DFS : a - b - d - e - c - f - g.

1. use BFS and DFS find the spanning tree for given graph

Sol :- BFS :-

1. The root node is a

(a) a

2. node a has two childs b and e.



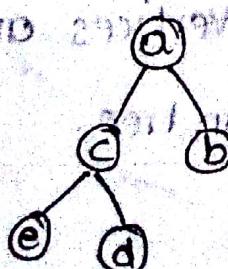
4. node e and d has child

3. node c has 2 childs d and e

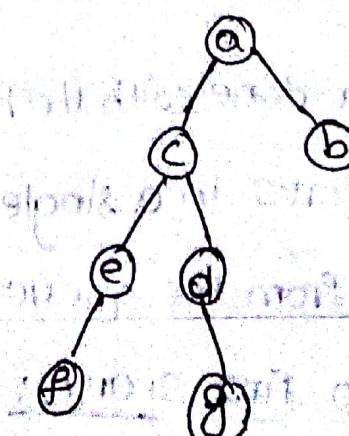
nodes f and g.

and b has child d but it will

be derived the cycle.

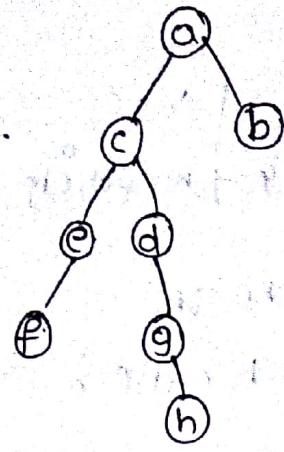


a - c - b - e - d



a - b - c - b - e - d - f - g

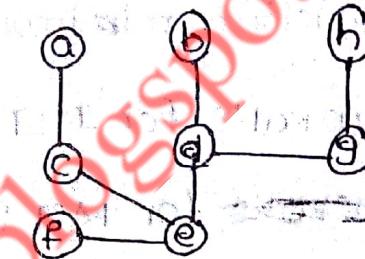
5. g has one child 'h'.
the BFS is a-c-b-e-d-f-g-h.



17-710
DFS:-

$v=a$	$v=c$	$v=e$	$v=f$	$v=d$
$w=\{c, b\}$	$w=\{e, d, a\}$	$w=\{f, d, c\}$	$w=\{e\}$	$w=\{e, g, b, c\}$
$v=g$	$v=h$	$v=b$		
$w=\{h, d\}$	$w=\{g\}$	$w=\{d, a\}$		

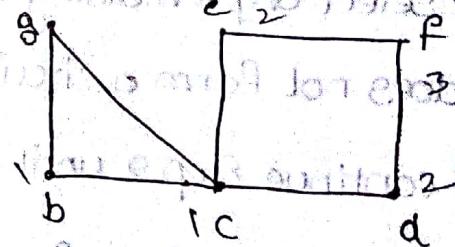
\therefore The DFS is a-c-e-f-d-g-h-b,



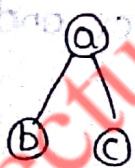
2. Determine BFS and DFS from the given graph

Sol:- BFS:-

1. The root node is a.

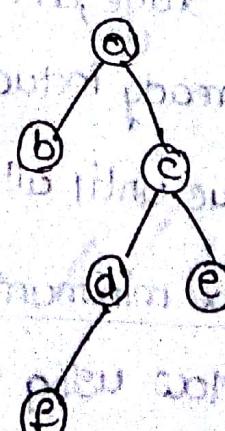
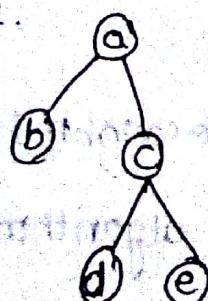


2. root node a has 2 childs b and c.



4. node d has one new child node f.

3. node b does not have one more new child, it has two child nodes d and e.



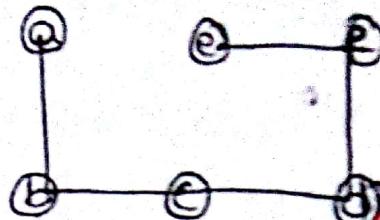
The BFS path is a-b-c-d-e-f

DFS:-

$V=a$	$V=b$	$V=c$	$V=d$	$V=f$
$W=\{b,c\}$	$W=\{a,c\}$	$W=\{d,e\}$	$W=\{f,c\}$	$W=\{d,e\}$

$$V=e \\ W=\{f,c\}$$

∴ the DFS is
a-b-c-d-f-e.



MST (minimum Spanning tree) :- A minimal spanning tree of G is a spanning tree with minimum weight. To construct a minimum spanning tree use following Techniques,

1. Kruskal's for MST

Prims

2. Prim's for MST

1. ~~for MST~~ for MST :- 1. Select any edge of minimum value that is not a loop this is the first edge of T .

2. Select any remaining edge of G having minimal value that does not form a circuit with the edges already included in T .

3. Continue Step 2 until tree contains $n-1$ edges.

Kruskal's

2. ~~algorithm~~ algorithm for MST :- 1. Select any vertex and choose the edge of minimum weight from G .

2. At each stage, choose the edge of smallest weight joining a vertex already included to vertex, not yet included.

3. Continue until all vertices are included.

1. Find the minimum spanning tree of the weighted graph given below using Kruskal's and Prim's algorithm.

Prims

Sol:- ~~for MST~~ for MST

edges $\{a,e\}$ $\{a,d\}$ $\{a,c\}$

Weights 3 3 3

$\{a,b\}$ $\{b,e\}$ $\{b,d\}$ $\{b,c\}$

4 3 1 2

$\{c,d\}$ $\{d,e\}$

3 2

1. a is root node

(a)

2. we have consider

$\{a,e\} = 3$



3. we have to consider

$\{e,d\} = 2$



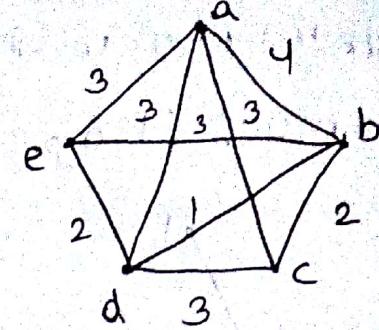
Kruskal's
~~Algo~~ forms:-

edges $\{a,e\}$ $\{a,d\}$ $\{a,c\}$ $\{a,b\}$ $\{b,e\}$ $\{b,d\}$ $\{b,c\}$

Weights 3 3 3 4 3 2 2

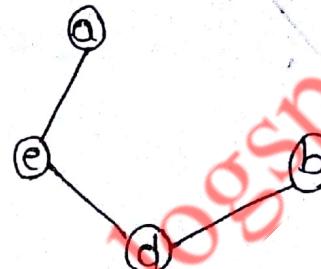
$\{c,d\}$ $\{d,e\}$

3 2



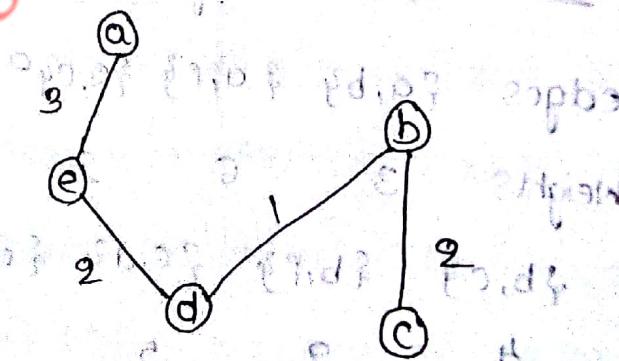
4. We have to consider

$\{d,b\} = 1$



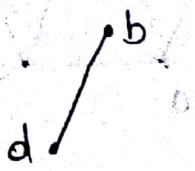
5. we have to consider

$\{b,c\} = 2$

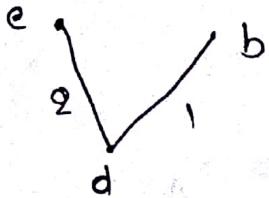


1. the minimum weight of

all nodes $\{b, d\}$



2. next $\{d, e\} = 2$



2. construct the minimum spanning tree using Kruskal's and Prim's algorithm.

Sol:-

~~Prim's~~ algorithm:-

edges $\{a, b\}$ $\{a, f\}$ $\{a, e\}$ $\{a, d\}$

Weights

3 6 2

$\{b, c\}$ $\{b, f\}$ $\{c, d\}$ $\{d, f\}$ $\{d, e\}$

4

2

5

1

4

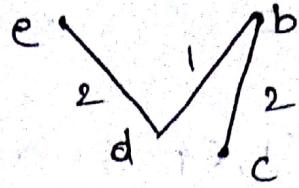
$\{f, e\}$

3

1. a is rootnode

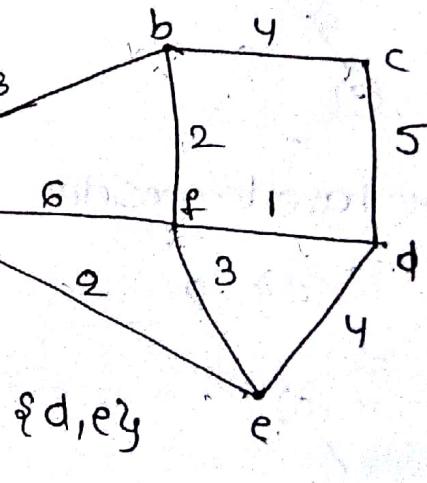
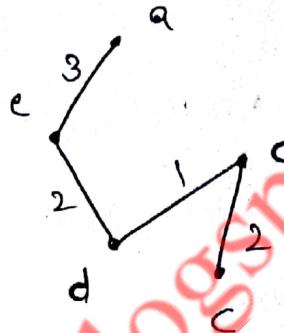
@

2. We have consider $\{a, e\} = 2$

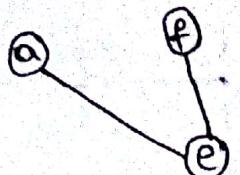


4. next minimum weight

$\{a, e\} = 2$

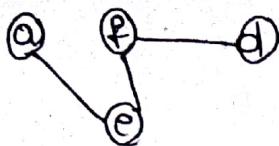


3. We have to consider $\{e_1, f_3\} = 3$

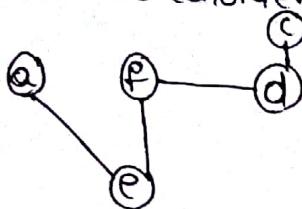


4. We have to consider

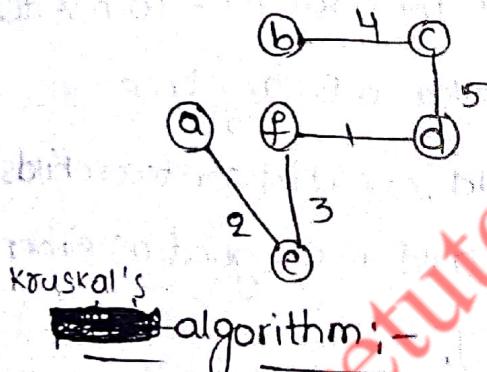
$$\{f_1, d\} = 1$$



5. We have to consider $\{d, e\} = 5$



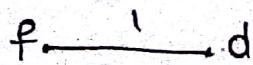
6. We have to consider $\{c, b\} = 4$



Kruskal's algorithm:-
edges $\{a, b\}$ $\{a, f\}$ $\{a, e\}$ $\{b, c\}$ $\{b, d\}$ $\{c, d\}$ $\{d, e\}$,
weights 3 6 2 4 2 5

1. the minimum weight of

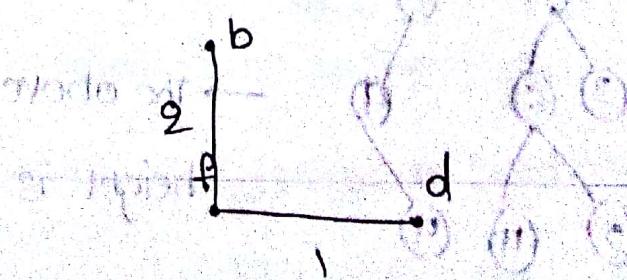
all nodes is $\{d, f\}$



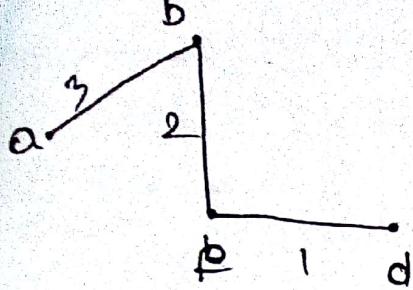
2. next $\{b, f\} = 2$

$\{d, e\}$. $\{f, e\}$: x1

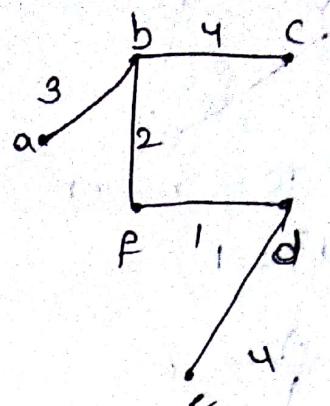
4, 3



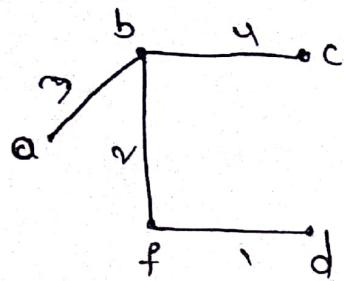
3. next of a, b = 3



5. next of d, e = 4.



4. next of b, c = 4



19-9-10

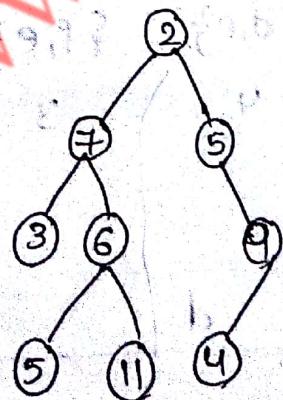
Binary tree: A rooted tree in which the children of each vertex are assigned a fixed ordering is called a Binary tree.

2. If either each vertex has no child, one child (or) two children.

3. If a tree has one child then that child is designed as either leftchild (or) rightchild [but not both].

4. If a vertex (or) node has two children then the first child is designed as leftchild other child is designed as right child.

Ex:-



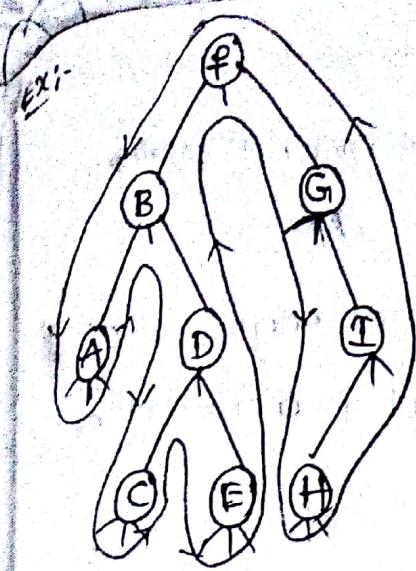
size is 9

height $h = 3$.

→ The above tree size is '9' and height is '3'.

The root node is '2'.

The child nodes are 3, 5, 11, 4.



preorder:- FBADCEGIH

Inorder:- ABCDEFIGHI

postorder:- ACEDBHIGf.

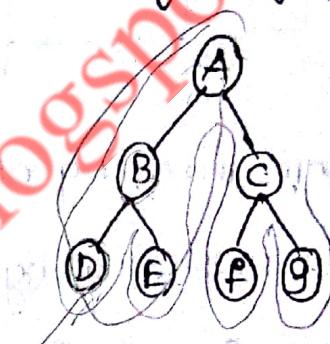
1. construct preorder, inorder and postorder from given graph.

sol:- preorder:-

ABDECFG

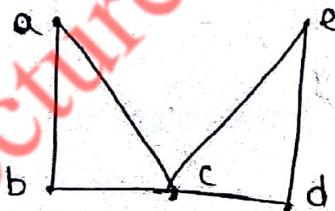
inorder:- DBEAFCG

postorder:- DEBFGCA.



cut vertex:- A cut vertex of a connected graph 'G' is a vertex, whose removal increases the number of components.

Ex:-

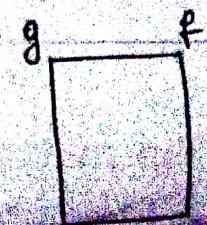


→ Here the vertex 'c' is a cut vertex.

→ If we are removing the vertex 'c'. It is dividing into two components {a, b} & {d, e}.

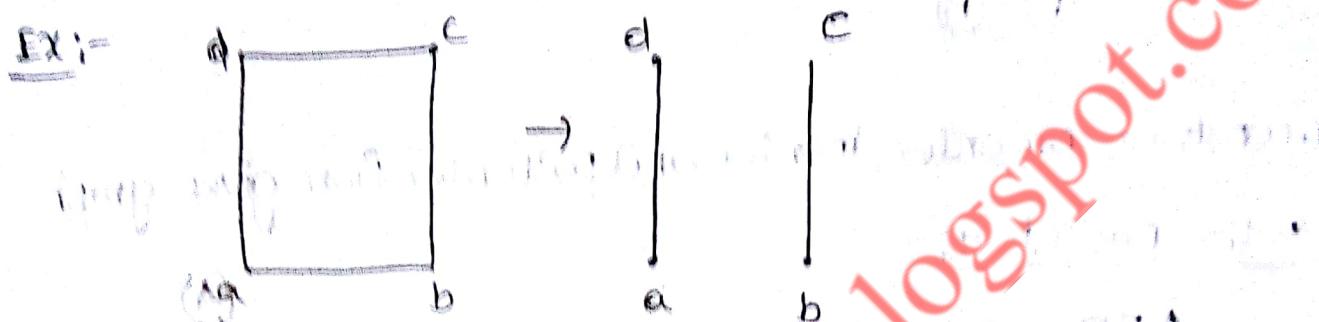
cut edge (Bridge):- A cut bridge of a connected graph 'G' is an edge which is removed increases the number of components.

Ex:-

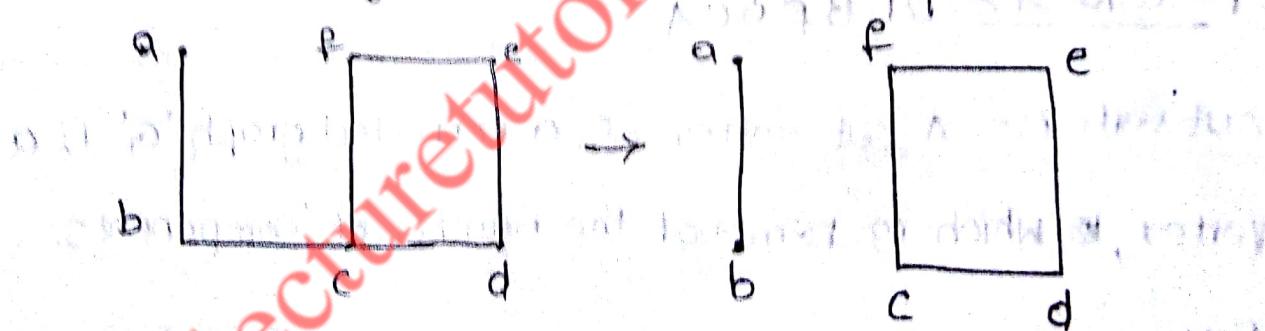


- Here (c-d) edge is a bridge
- If we are removing (c-d) edge, the graph is divided into two components.

Cut Set: The set of all minimum number of edges of G' which is removal (or) disconnect a graph is a cut set of G .



- Here removing of two edges $\{(a-b), (c-d)\}$.
- The graph dividing into two components



- Here the cut sets is $\{(b-c)\}$