29-8-16

s. lakshmi ...etha

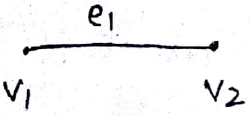
VI. GRAPH THEORY

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 a another set whose elements are called edges (lines).

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

$G = (V, E)$ (or) $G(V, E)$.

Ex:-



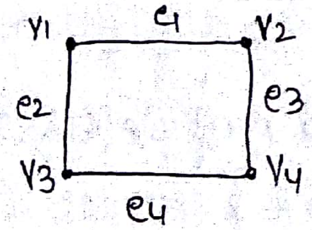
Here $V = \{v_1, v_2\}$, $E = \{e_1\}$

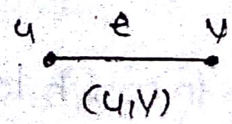
Note:- 1. If an edge $e \in E$ is associated with an ordered pair (u, v) where $(u, v) \in V$.

2. e(edge) is connected to u and v are 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.



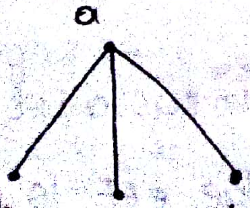


Here e has two adjacent vertices u and v.

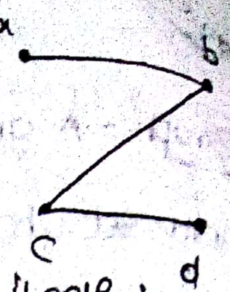
In the above graph v_1 has two adjacent edges e_1 and e_2 and v_1 denoted as incident vertex.

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

Sol:-



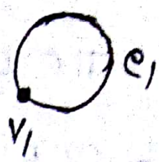
2. construct the vertices and edges from given graph.



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

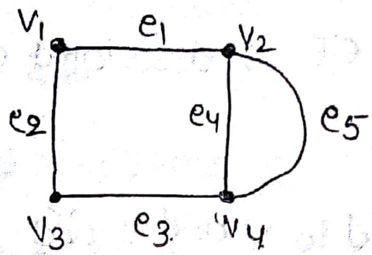
$V = \{a, b, c, d\}$ $E = \{(a,b)(b,c)(c,d)\}$

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



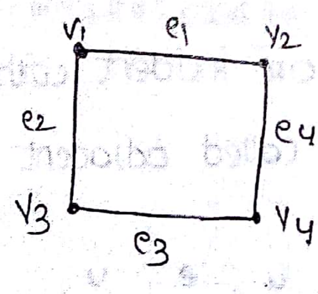
31-8-16

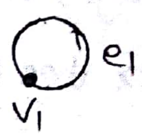
Multigraph:- If more than one line (edge) joining between two vertices are allowed in a graph then the graph is called multigraph. Ex:-



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

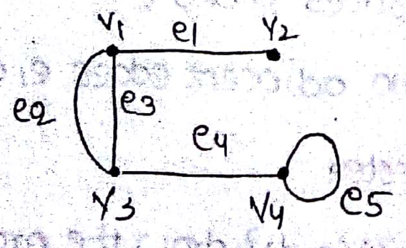
Ex:-





Psuedo graph:- A graph in which loops and multiple edges are allowed is called psuedo graph.

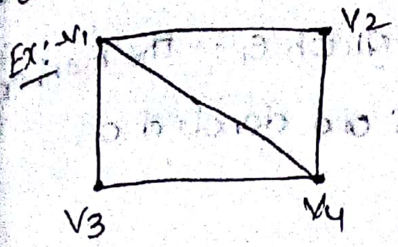
Ex:-



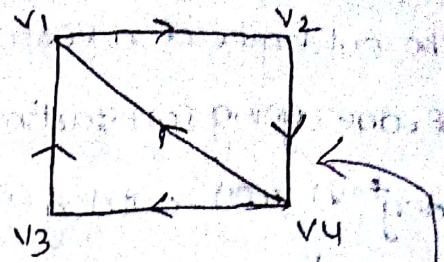
Directed and undirected graph:-

undirected graph:- An undirected graph G 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))

Ex:-



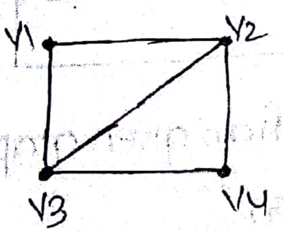
Ex:-



directed graph:- A directed graph G 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 is the no. of edges incident with it. The degree of that vertex denoted as deg(V) (or) d(V).

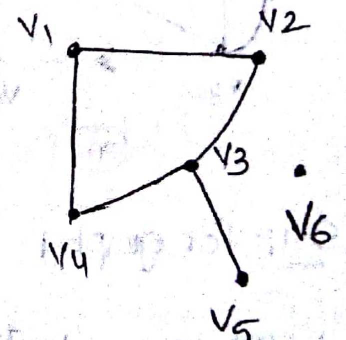
Ex:-



deg(v_1) = 2 deg(v_3) = 3

deg(v_2) = 3 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.

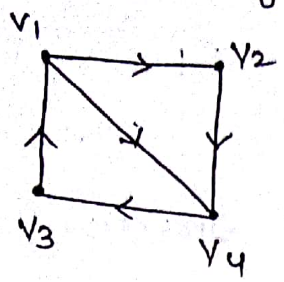
1/9/16

*In-degree and out-degree on directed graphs:-

The In degree of a vertex V of a directed graph G is the number of edges receiving (or) ending (or) coming at V and denoted as $\deg^-(V)$ (or) Indeg(V).

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

Ex:-



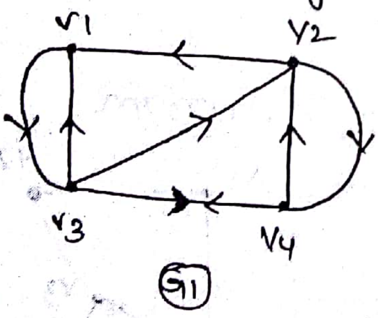
Indegree	out-degree
$\deg^-(V_1) = 1$	$\deg^+(V_1) = 2$
$\deg^-(V_2) = 1$	$\deg^+(V_2) = 1$
$\deg^-(V_3) = 1$	$\deg^+(V_3) = 1$
$\deg^-(V_4) = 2$	$\deg^+(V_4) = 1$

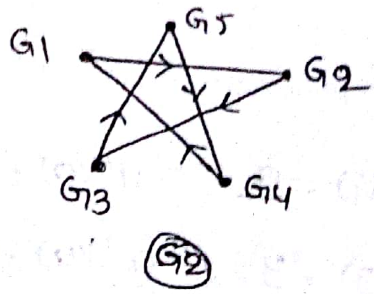
Note:- If $G = (V, e)$ is a directed graph with edge e then

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

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

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

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

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

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

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

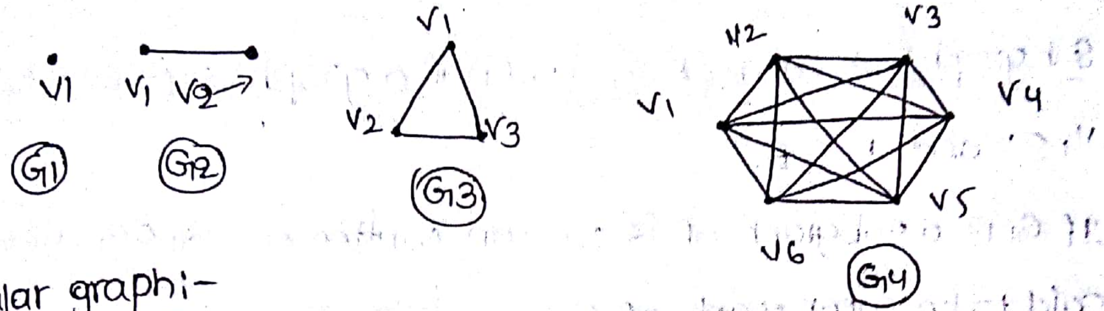
$\deg^-(G_5) = 1$ $\quad \deg^+(G_5) = 1$

Null graph:- A graph G in which node (or) vertex is isolated node is called a null graph. the vertex V has '0' edges.

Ex:- v_1 v_2

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

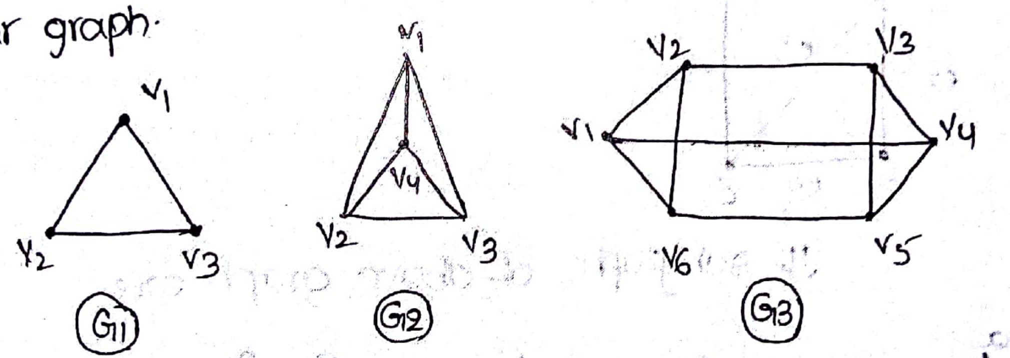
Ex:-



Regular graph:-

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

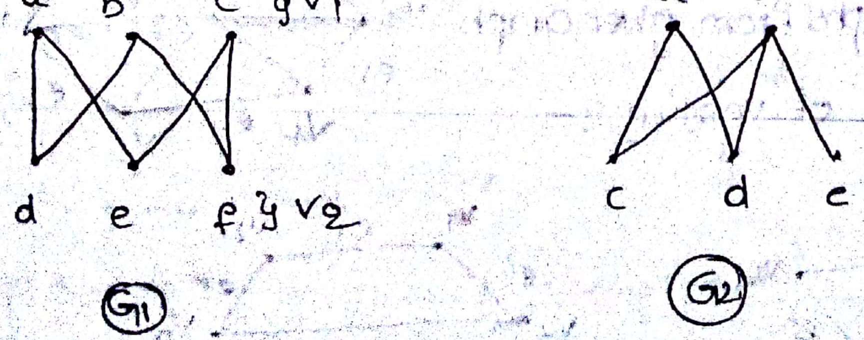
Ex:-

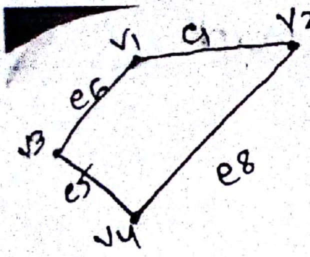


2-9-

***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.

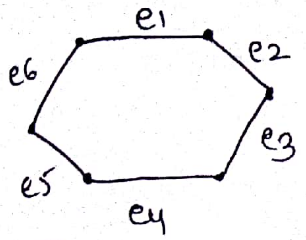
Ex:-

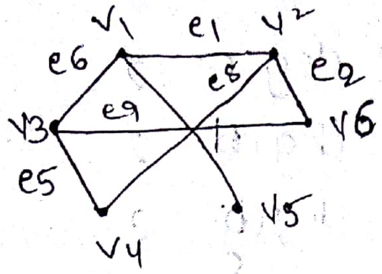




(V5, V6)

elimination of edges:-





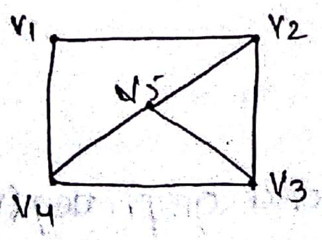
4-9-[illegible]

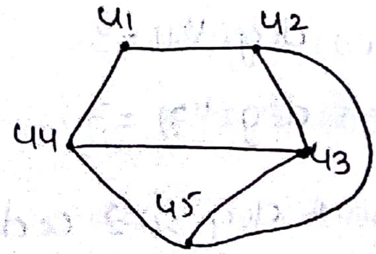
Isomorphism:- Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be Isomorphic if there exists a Bijection $f: V_1 \rightarrow V_2$ such that $(u_1, v_1) \in E_1$ (or) adjacent vertices in G_1. $(u_2, v_2) \in E_2$ are adjacent vertices in G_2.

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).

Ex:-





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

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

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

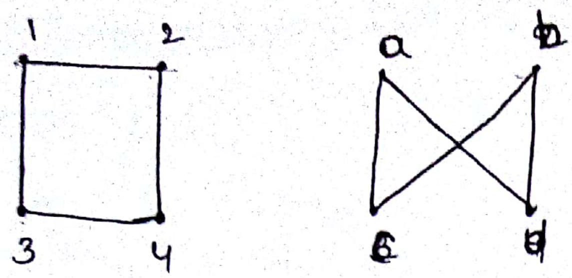
$\deg(v_3) = 3 \Leftrightarrow \deg(u_3) = 3$

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

$\therefore$ The given two graphs are in isomorphism.

1. Show that the following graphs G and G' are isomorphic.

(a)



Sol:- deg(1) = 2 ⟺ deg(a) = 2

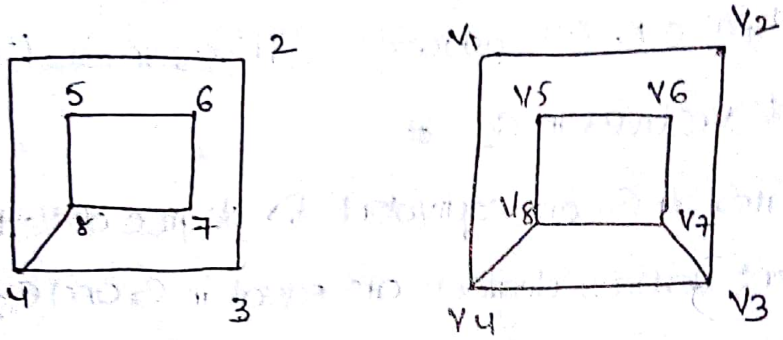
deg(2) = 2 ⟺ deg(d) = 2

deg(3) = 2 ⟺ ~~deg(b) = 2~~ deg(c) = 2

deg(4) = 2 ⟺ ~~deg(d) = 2~~ deg(b) = 2

∴ The given two graphs are isomorphism.

(b)



deg(1) = 2 ⟺ deg(V1) = 2

deg(2) = 2 ⟺ deg(V2) = 2

deg(4) = 3 ⟺ deg(V4) = 3

deg(3) = 2 ⟺ deg(V3) = 3.   Isomorphic

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

∴ These two graphs are not Isomorphic.

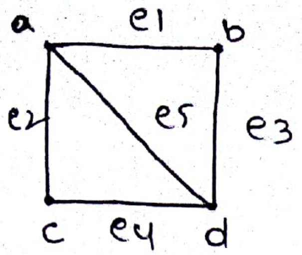
3) Matrix representation of a graph:- matrix representation of a graph has 2 types. 1. Adjacence matrix. 2. Incidence

1. Adjacence matrix:-

Let $G = (V, E)$ be a simple graph with n vertices ordered from V_1 to V_2 then the adjacence matrix $A_m = [a_{ij}]_{n\times n}$ of G is an $n\times n$ symmetric matrix defined by $A_m = [a_{ij}]_{n\times n}$ of an $n\times n$

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

Ex:-



$$Am = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 1 \\ b & 1 & 0 & 0 & 1 \\ c & 1 & 0 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{array}$$

8-9-10

2. Incidence matrix :-

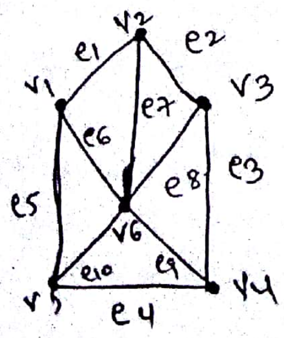
Let G be a graph with n vertices $V = \{v_1, v_2, v_3 \text{ ---- } v_n\}$ and $E = \{e_1, e_2 \text{ ------ } e_m\}$ define $n \times m$ matrix $Im = [a_{ij}]_{n \times m}$

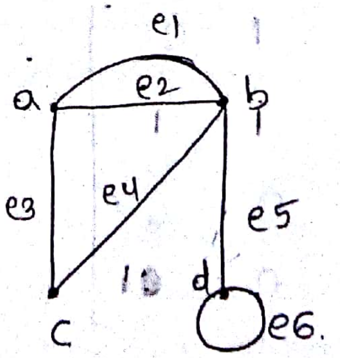
Where $a_{ij} = \begin{cases} 1 & \text{When } v_i \text{ is Incident with } e_j \\ 0 & \text{otherwise} \end{cases}$

Ex:-

$$Am = \begin{array}{c|ccccc} & e_1 & e_2 & e_3 & e_4 & e_5 \\ \hline a & 1 & 1 & 0 & 0 & 1 \\ b & 1 & 0 & 1 & 0 & 0 \\ c & 0 & 1 & 0 & 1 & 0 \\ d & 0 & 0 & 1 & 1 & 1 \end{array}$$

1. find the adjacence and Incidence matrix from given graphs





Sol:- The adjacent matrix from first graph is Am =

The Incidence " " " " is Im =

$$
A_m = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

$$
I_m = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\end{array}
$$

The adjacent matrix to Second graph

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

The Incidence matrix to second graph is

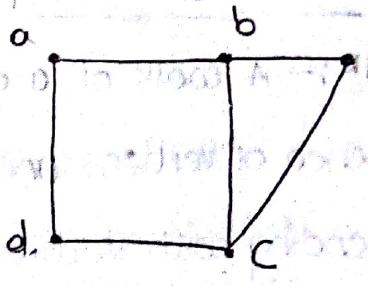
$$\text{Im} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Draw the graph represented by the adjacence matrix.

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

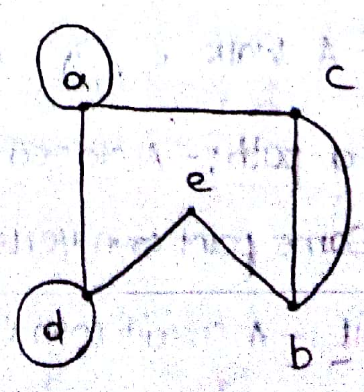
Sol:- The given adjacence matrix is

Let $A_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$



The given adjacence matrix is

Let $A_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$

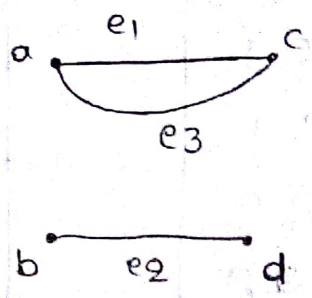


3. Draw the graph from given Incidence matrix

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

Sol:- The given Incidence matrix is

$$I_G = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$



9-9-19

Paths and circuits:-

1. Walk:- A walk of a graph G is defined as an alternating Sequence of vertices and edges. $v_0 e_0 v_1 e_1 \ldots\ldots e_n v_n$. 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.

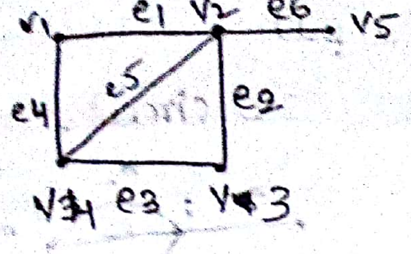
2. Trail:- A walk is called a Trail if all its edges are distinct.

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).

Ex:-



1. $v_1 e_1 v_2 e_2 v_3$

2. $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_5 v_2 e_2 v_3$

are walks.

3. $v_1 e_1 v_2 e_5 v_4 e_3 v_3 e_2 v_2 e_6 v_5$ is a trail.

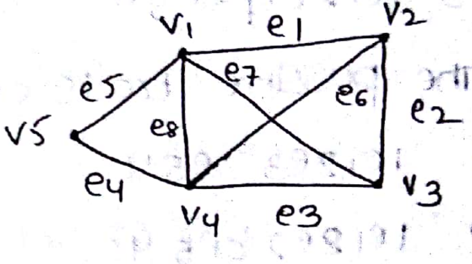
4. $v_1 e_4 v_4 e_3 v_3 e_2 v_2 e_6 v_5$ is a path.

5. $v_1 e_1 v_2 e_5 v_4 e_3 v_3 e_2 v_2 e_1 v_1$ is a closed path

6. $v_1 e_4 v_4 e_5 v_2 e_1 v_1$

7. $v_1 e_4 v_4 e_3 v_3 e_2 v_2 e_1 v_1$ are circuits.

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



1. $v_1 e_1 v_2 e_6 v_4 e_3 v_3 e_2 v_2$

2. $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5$

3. $v_1 e_8 v_4 e_3 v_3 e_7 v_1 e_8 v_4$

4. $v_5 e_5 v_1 e_8 v_4 e_3 v_3 e_2 v_2 e_6 v_4 e_4 v_5$

5. $v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_1 e_1 v_2$

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.

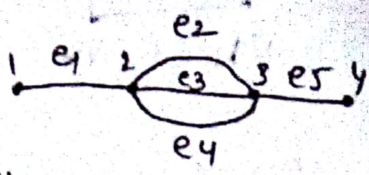
Starting vertex v_1 ending vertex v_4, so it is " " circuit

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_2 is repeated, so it is not a path. Starting vertex V_2, ending vertex V_2, So it is a circuit.

2. 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. $1e_12e_33e_54$
2. $1e_12e_23e_54$
3. $1e_12e_43e_54$

The possible trails are from 1 to 4 is

1. $1e_12e_33e_54$
2. $1e_12e_23e_54$
3. $1e_12e_43e_54$
4. $1e_12e_23e_32e_43e_54$
5. $1e_12e_23e_42e_33e_54$
6. $1e_12e_43e_22e_33e_54$
7. $1e_12e_43e_32e_23e_54$
8. $1e_12e_33e_42e_23e_54$
9. $1e_12e_33e_22e_43e_54$.

*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:-

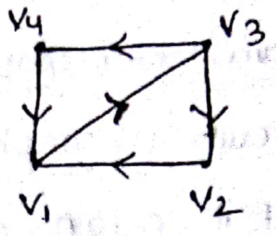
1. A path in a Graph G is called an Euler path if it includes every edge exactly once (distinct edges).
2. visit all vertices at least once.

Note:- 1. If G is a graph in which the degree of every vertex is even then it is possible to construct Euler circuit.

2. The graph G is a Euler path if atleast one degree of vertex is even.

3. If the given graph G 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

$\deg^-(V_1) = 2$, $\deg^-(V_2) = 1$, $\deg^-(V_3) = 1$, $\deg^-(V_4) = 1$

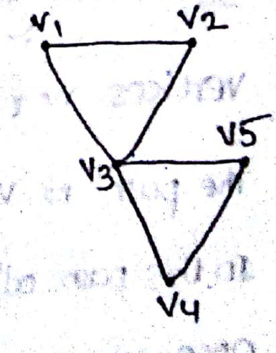
outdegree to every vertex is

$\deg^+(V_1) = 1$, $\deg^+(V_2) = 1$, $\deg^+(V_3) = 2$, $\deg^+(V_4) = 1$

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

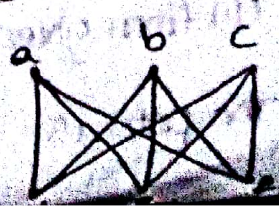
vertices $V = \{V_1, V_2, V_3, V_4, V_5\}$.

$\deg(V_1) = 2$, $\deg(V_2) = 2$, $\deg(V_3) = 4$

$\deg(V_4) = 2$, $\deg(V_5) = 2$

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.

13-9-18

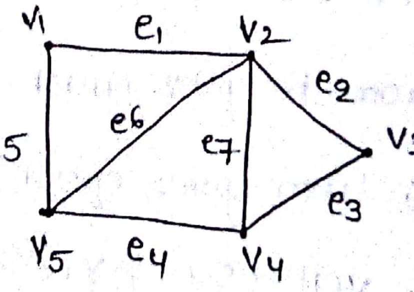
Hamaltonian graph:- A circuit in a graph G is called Hamoltonia circuit (or) graph.

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

Hamaltonian path:- A Hamaltonian 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 Hamaltonian circuit or path.

(i)



Sol:- (i) from given graph

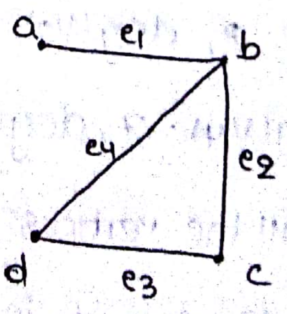
Vertices $V = \{v_1, v_2, v_3, v_4, v_5\}$

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.

∴ It is Hamaltonian circuit.

(ii)



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

The path is a-b-c-d.

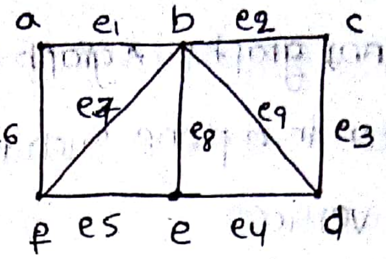
In this path all vertices visited exactly once and starting and ending vertices are distinct.

$\therefore$ It is a Hamaltonian path.

2. The given graph which is Hamaltonian circuit (or) Euler circuit

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

(i)

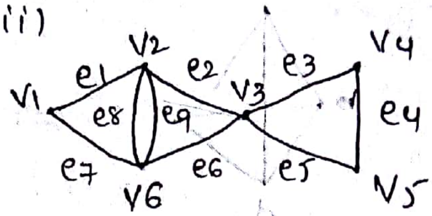


$V = \{a, b, c, d, e, f\}$

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

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

(ii)



$\therefore$ It is a Hamaltonian circuit.

case2:- from 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 $V_1 - V_2 - V_6 - V_3 - V_4 - V_5$.

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

$\therefore$ It is not a Hamaltonian circuit.

case2:- from given graph degrees to all vertices

$deg(V_1) = 2$, $deg(V_2) = 4$, $deg(V_3) = 4$, $deg(V_4) = 2$

$deg(V_5) = 2$, $deg(V_6) = 4$

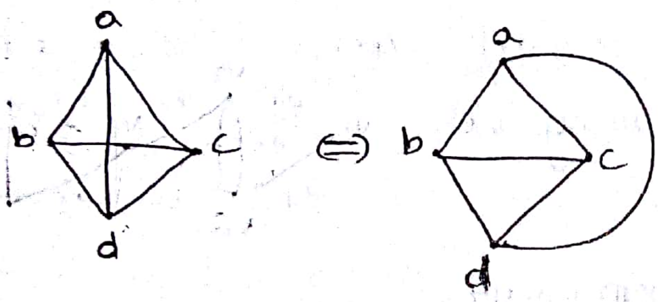
The path is $V_1 - V_2 - V_3 - V_4 - V_5 - V_3 - V_6 - V_2 - V_6 - V_1$.

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

∴ It is Euler circuit

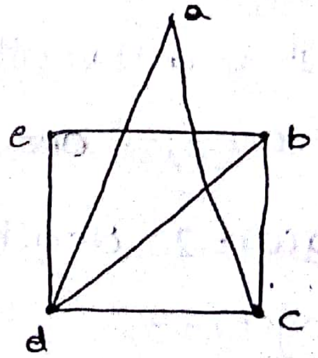
Planar graph:- A graph G is called a planar graph if it can be drawn in a plane such that no two edges intersect except at the vertices.

Ex:-

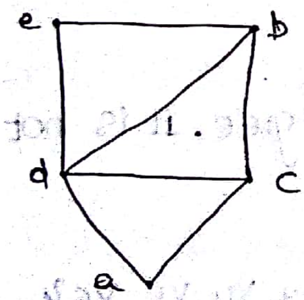


15-9-10

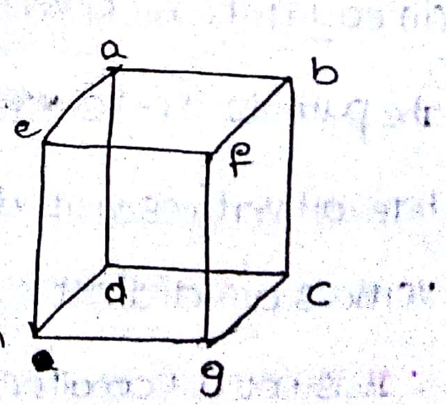
1. Determine the graph is it planar (or) not.



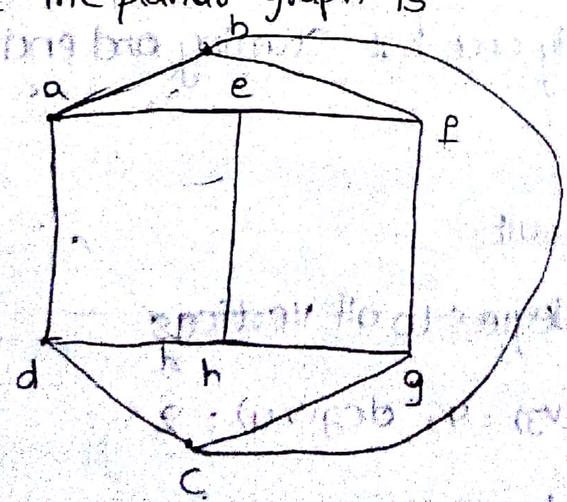
Sol:- The planar graph is



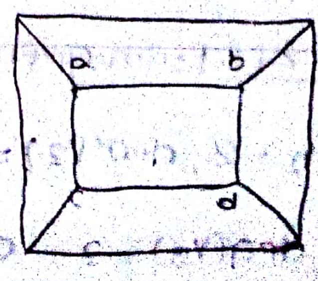
2. The given graph is planar (or) not.



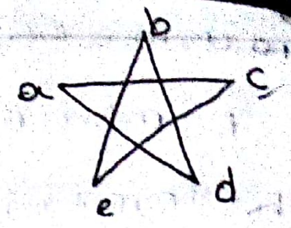
Sol:- The planar graph is



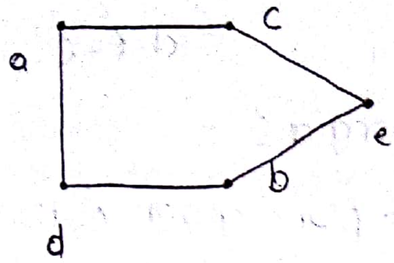
(or)



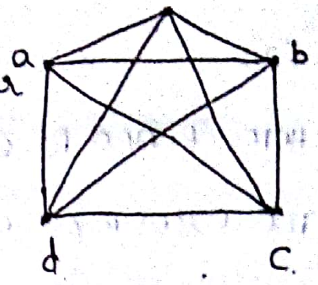
3. The given two graphs are planar or not.



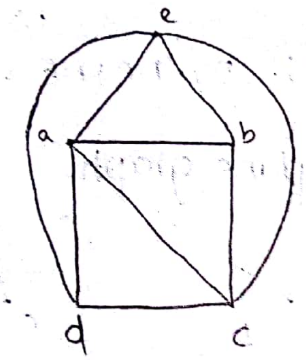
sol:- The planar graph is



It is a planar graph from given graph.



The planar graph is



e, d, b. c

d, a, e, c

a, b, d, e.

a, c, b, e

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.

~~Graph colouring~~ and ~~covering~~:-

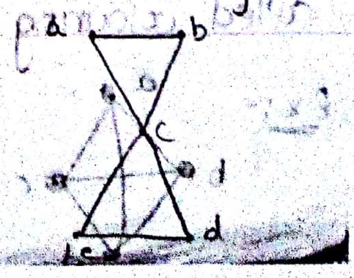
characteristics:- $n_v - n_e + n_f$

where n_v = number of vertices in a graph.

n_e = " " edges

n_f = " " faces (or) regions

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



(i)

no. of vertices $n_v = 5$

" " edges $n_e = 6$

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

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

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

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

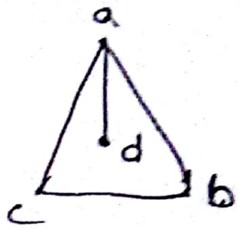
Here R_1 and R_2 are interior regions.

The exterior region R_3 = The plane graph outside path a-b-c-d-e-c-a.

$\therefore nf = 3$.

The characteristics to given graph $nv - ne + nf = 5-6+3 = 2$.

2. construct the characteristics to plane graph.



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

no. of edges $ne = 4$

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

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

Here R_1 is interior region

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

$\therefore nf = 2$

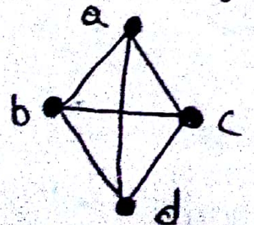
The characteristics to given graph $nv - ne + nf = 4-4+2 = 2$.

8-9-16

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) = Red

f(b) = Blue

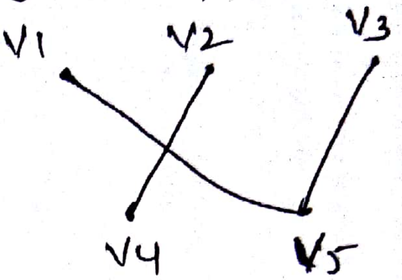
f(c) = pink

f(d) = Yellow

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

$\therefore \chi(G) = 4$

1. Determine the chromatic number of given Bipartiate graph.



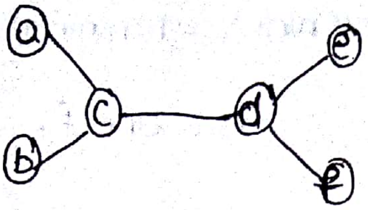
Sol:- $\therefore f(V1) = R,\ f(V2) = R,\ f(V3) = R$

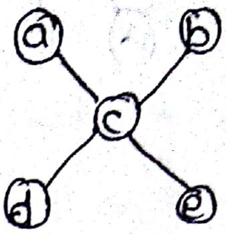
$f(V4) = B,\ f(V5) = B$

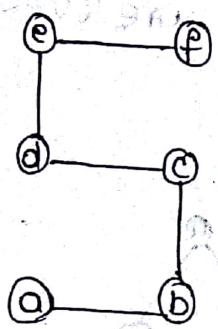
$\chi(G) = 2$

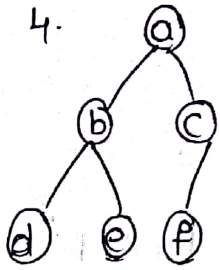
16-9-19 Trees

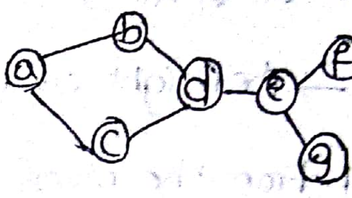
A tree is a simple graph G such that there is a unique simple undirected path between each pair of vertices in G.

Ex:- 1. 

2. 

3. 

4. 

5. 

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

→ tree is denoted as 'T'.

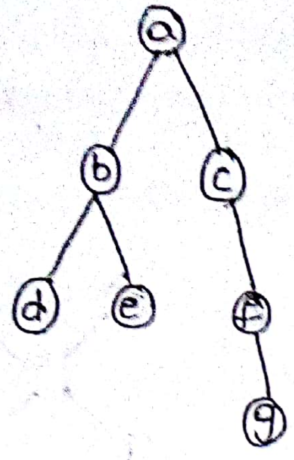
rooted tree:- A rooted tree is a tree in which a particular vertex is designed as the root (starting node (or) vertex).

→ 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 ($v_0, e_1, v_1, e_2 - - - e_n v_n$) 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.

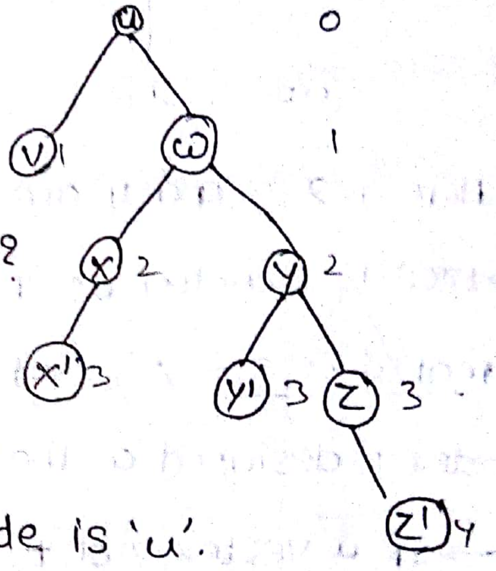
Here child nodes are d, e, and g.

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

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

→ The height of tree is '3'.

1. 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 ω 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, ω, x, y, z, x', y', z'

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

The Interval vertices are u, ω, x, y, z.

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

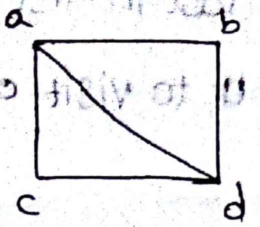
(iv) The childs of W and z are x', y', z' and z'.

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.

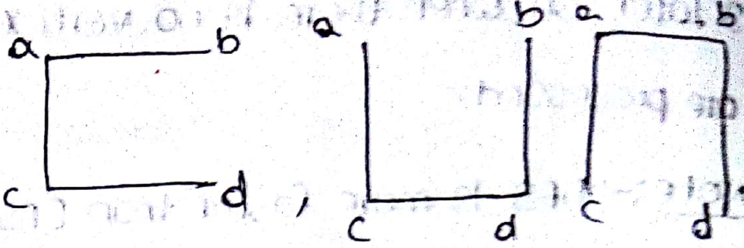
If G is a connected graph with n vertices and m edges, a spanning tree of G must have n-1 edges.

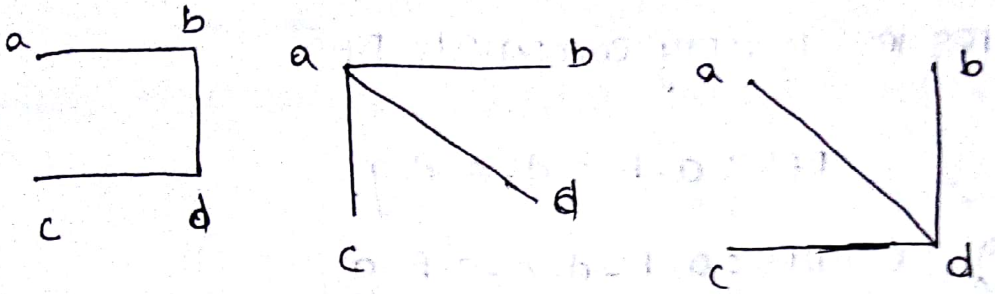
$\boxed{m = n-1}$

Ex:-



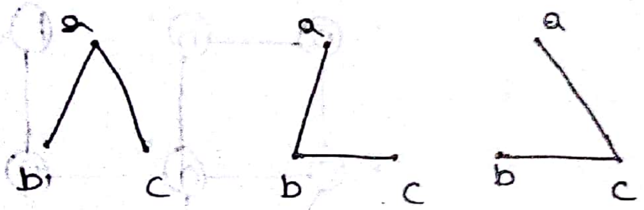
$\Rightarrow$ The spanning trees are





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

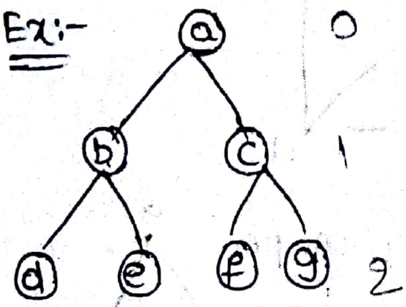
2. DFS can be done with the help of stack (last in first out).

3. Later on when there is no vertex further to visit all vertices are processed.

<u>Note</u>:- 1. DFS is more faster than BFS.

2. DFS requires less memory compare to BFS.

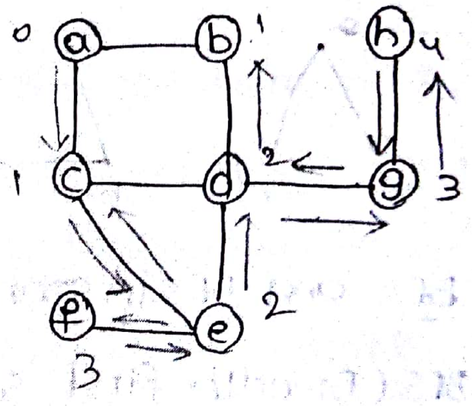
<u>Ex</u>:-



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

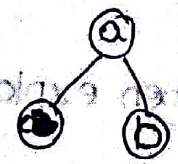


<u>Sol</u>:- <u>BFS</u>:-

1. The root node is a

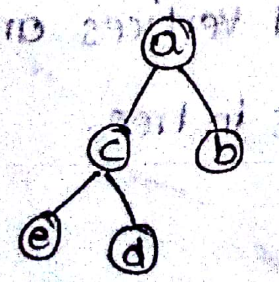
a a

2. node a has two childs b and c.



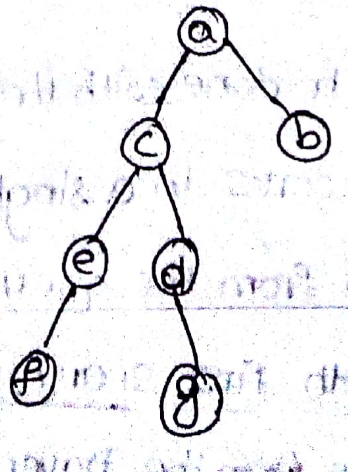
a-c-b

3. node c has 2 childs d and e and b has child d But it will be derived the cycle.



a-c-b-e-d

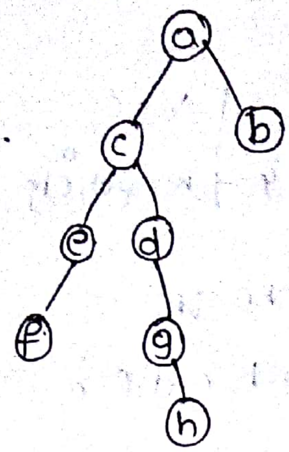
4. node e and d has child nodes f and g.



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.

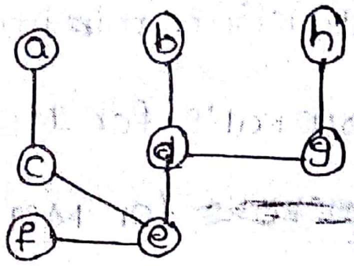


17-7-10

DFS:-

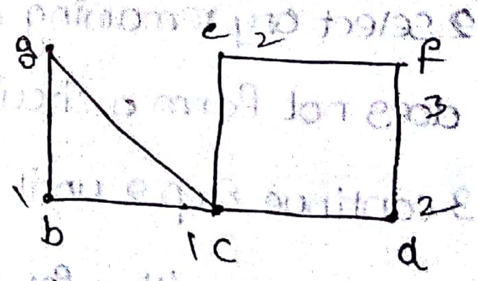
v=a W={c,b}	v=c ω={e,d,a}	v=e ω={f,d,c}	v=f ω={e}	v=d ω={e,g,b,c}

v=g ω={h,d}	v=h h={g}	v=b W={d,a}



∴ The DFS is a-c-e-f-d-g-h-b.

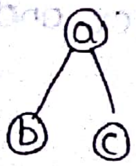
2. Determine BFS and DFS from the given graph



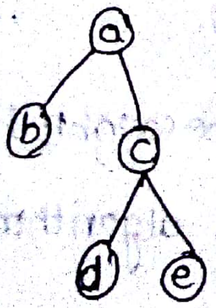
Sol:- BFS:-

1. The root node is a. (a)

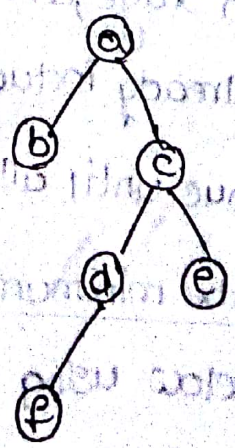
2. root node a has 2 childs b and c.



3. node b does not has one more newchild, c has two child nodes d and e.



4. node d has one new child node f.



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

DFS:-

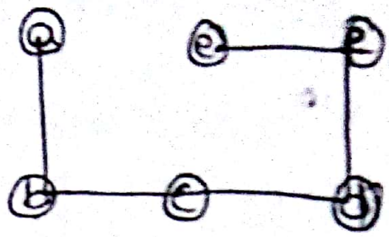
v = a	v = b	v = c	v = d	v = f
W = {b, c}	W = {a, c}	W = {d, e}	W = {f, c}	W = {d, e}

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 constructing minimum spanning tree use following Techniques.

1. Kruskal's for MST   2. prim's for MST.

1. Prim's 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.

2. Kruskal's algorithm for MST:- 1. select any vertex and choose the edge find minimum weight from G.

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

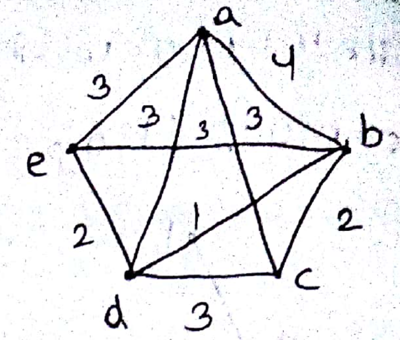
3. continue until all vertices are included.

1. find the minimum spanning tree of the weighted graph given below using kryskal's and prim's algorithm.

Sol:- Prim's 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

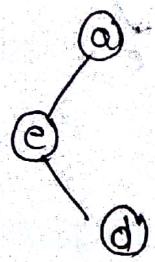
a

2. we have consider

{a,e} = 3

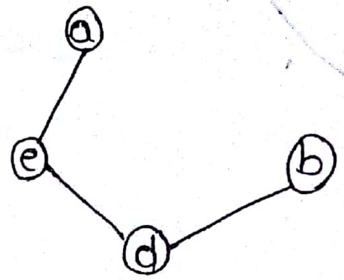
3. We have to consider

{e,d} = 2



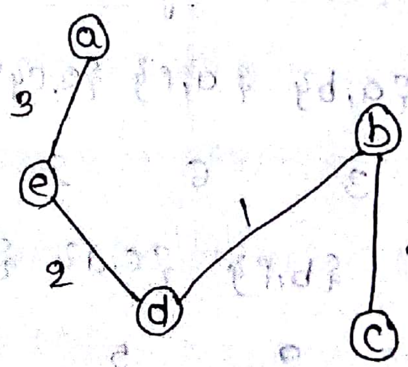
4. We have to consider

{d,b} = 1



5. We have to consider

{b,c} = 2



Kroushkal's

forms:-

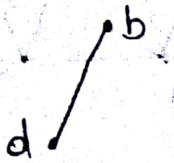
edges {a,e} {a,d} {a,c} {a,b} {b,e} {b,d} {b,c}

Weights 3 3 3 4 3 1 2

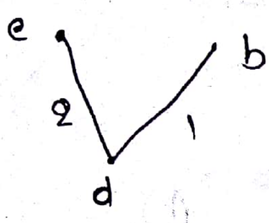
{c,d} {d,e}

3 2

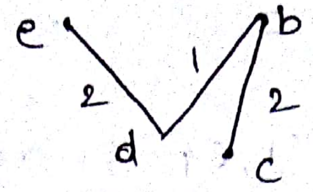
1. The minimum weight of all nodes {b, d}



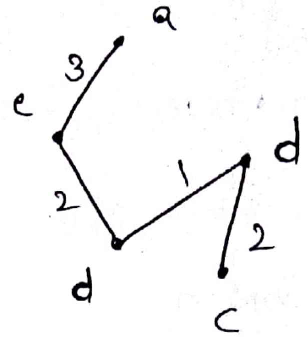
2. next {d, e} = 2



3. next minimum weight {b, c} = 2

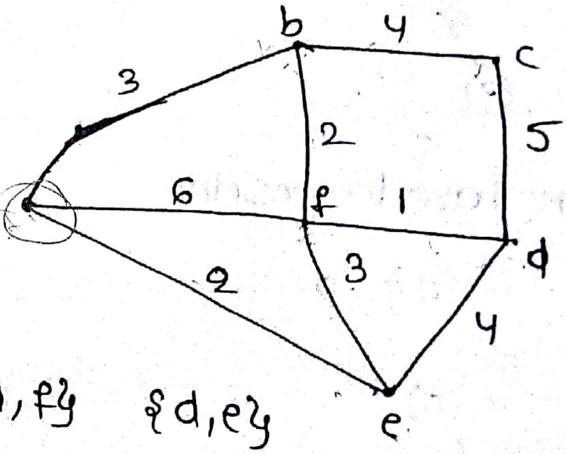


4. next minimum weight {a, e} = 3



2. construct the minimum spanning tree using krushkal's and prim's algorithm.

Sol:- Prim's algorithm:-



edges	{a, b}	{a, f}	{a, e}
weights	3	6	2

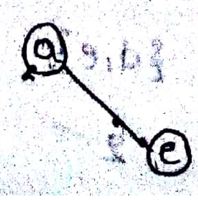
{b, c}	{b, f}	{c, d}	{d, f}	{d, e}
4	2	5	1	4

{f, e}

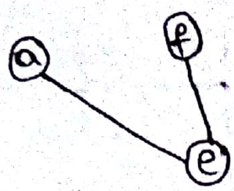
3

1. a is rootnode

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

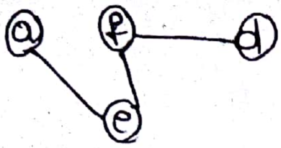


3. We have to consider {e, f} = 3

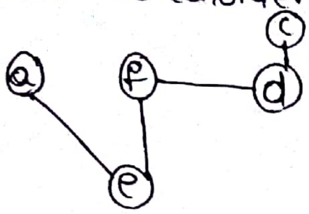


4. We have to consider

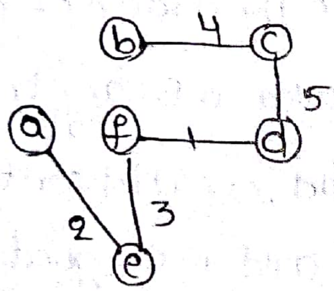
{f, d} = 1



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, f} {c, d} {d, f}

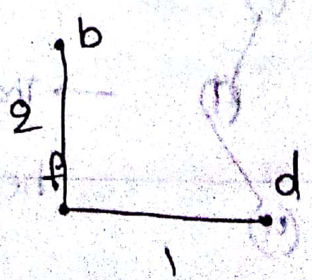
weights 3 6 2 4 2 5 1

{d, e} {f, e}

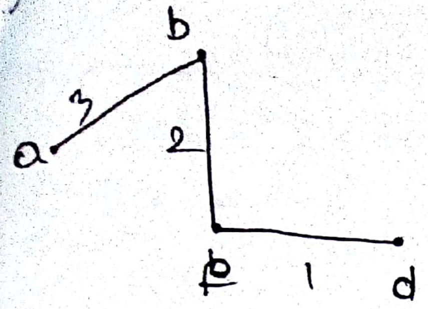
4 3

1. The minimum weight of all nodes is {d, f}

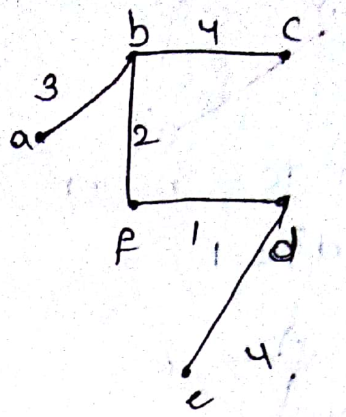
2. next {b, f} = 2



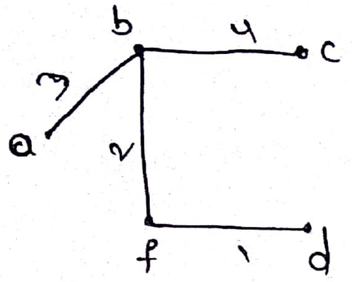
3. next {a,b} = 3



5. next{d,e} = 4.



4. next {b,c} = 4



19-9-1[illegible]

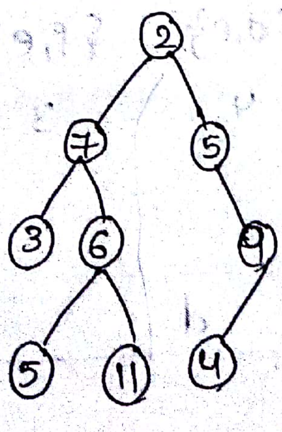
Binary tree:- 1. A rooted tree in which the children of each vertex are assigned a fixed ordering is called a Binary tree.

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

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 otherchild is designed as right child.

Ex:-



size is 9

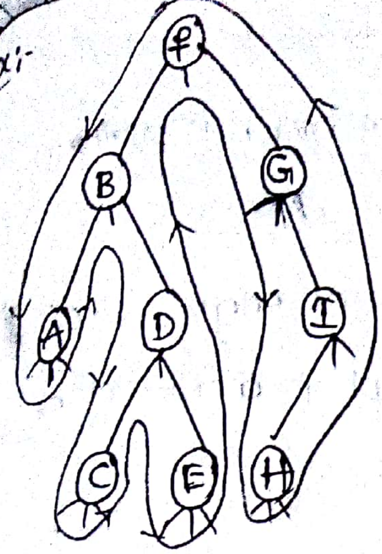
height h = 3.

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

The root node is '2'.

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

Ex:-

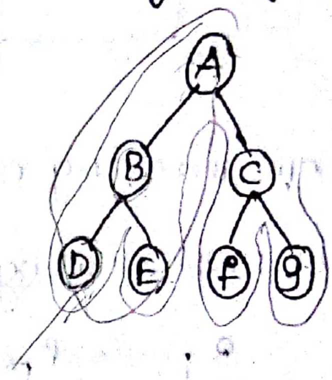


preorder:- FBADCEGIH

Inorder:- ABCDEFGHI

postorder:- ACEDBHIGF.

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



sol:- preorder:-

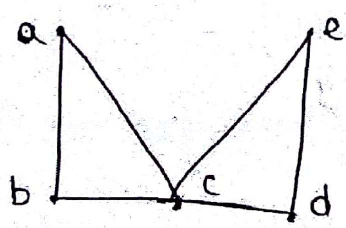
ABDECFg

Inorder:- ~~ABDECFg~~ DBEAFCg

postorder:- DEBFgCA.

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

Ex:-

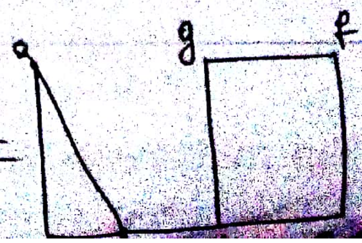


→ Here the vertex 'c' is a cut vertex.

→ 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 of 'G' is an edge which is removed Increases the number of components.

Ex:-

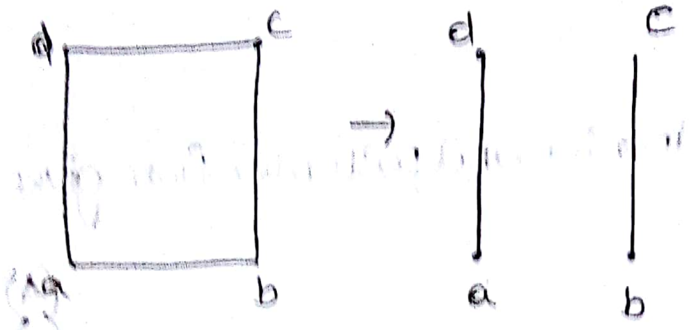


→ Here (c-d) edge cutedge (or) Bridge

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

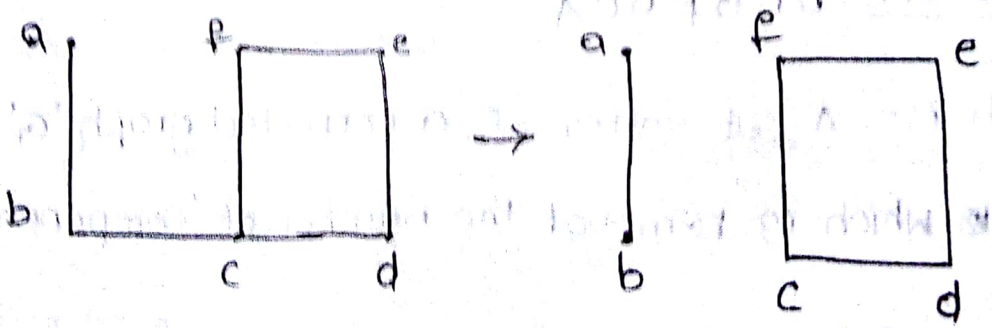
Cut set :- The set of all minimum number of edges of 'G' which is removal (or) disconnect a graph is a cut set of 'G'.

Ex:-



→ Here removing of two edges {(a-b), (c-d)}.

→ The graph dividing into two components



→ Here the cut sets is {(b-c)}