UNIT – 5

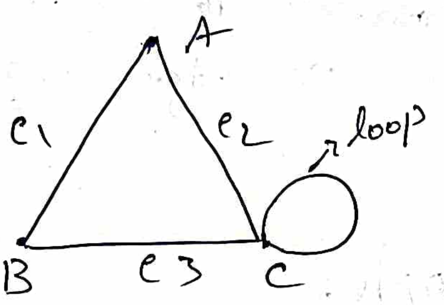
Graph Theory

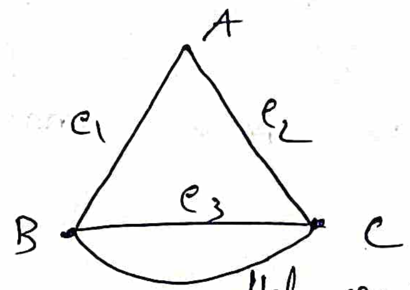
Introduction:

A graph is a collection of vertices and edges represented as $G=(V,E)$ where

V = vertex / node / point

E = Edge / line / curve





parallel graph

→ A, B, C be the ~~edges~~ vertex

→ e_1, e_2, e_3 be the edges.

→ If a edge starts and ends at the same vertex then it is called loop

→ If 2 edges has same starting and ending vertex then it is called parallel edges.

→ A graph with has loops and parallel edges ~~are~~ then called multiple graph

→ A graph does not contain loop and parallel edges then it is called <u>simple graph</u>

Eg :- △ → simple graph

Note :- In a simple graph with 'n' vertices there can be atmost $\frac{n(n-1)}{2}$ edges

Eg :- can we have graph with 6 vertices and 16 edges

$n = 6$,

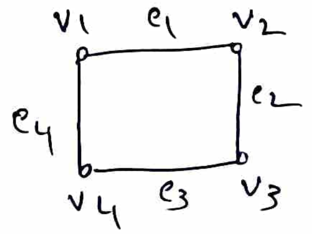
have max edges $\frac{n(n-1)}{2} = \frac{6(6-1)}{2} = \frac{6\times5}{2} = 15$ not

so not possible 16 edges.

so have max 15 edges only.

Types of graph.

Adjacent Nodes :- There is a connection between two vertex they are v_1v_2, v_2v_3, v_3v_4, v_1v_4 adjacent nodes.



Incident edges :- An edge which joins two notes/vertex is called as incident edge.

e_1 is incident edge on $v_1 \& v_2$

e_2 " " " " $v_2 \& v_3$

e_3 " " " " $v_3 \& v_4$

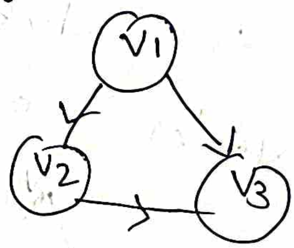
e_4 " " " " $v_1 \& v_4$

Directed edge :-

An edge which contain some directions is called directed edge i.e →

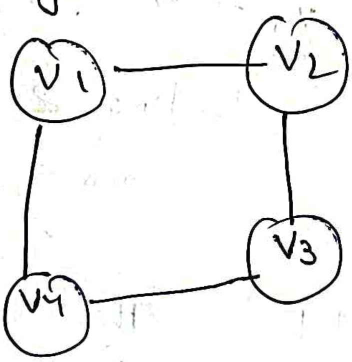
$G = (V, E)$

Directed graph :- If every edge in the graph is directed edge then we call it is directed graph



(V_1, V_3), (V_2, V_3), (V_1, V_2)

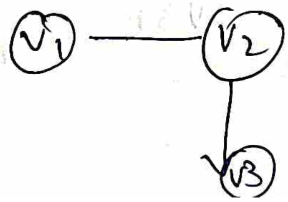
undirected edge :- In a graph $G=(V,E)$ An edge is associated with an unorderd pair of vertices then it is called as undirected edge



$(V_1 V_2)$ is undirected edge

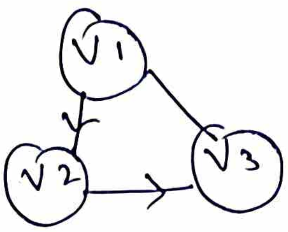
(V_2, V_1) means no direction

undirected graph :- If every edge in the graph is undirected edge then the graph is called undirected graph

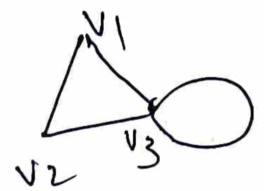


do not contain directions (V_1, V_2) $(V_2 V_1)$ $(V_3 V_1)$

Mixed graph :- If some edges have directions and some not have directions then it is called mixed graph



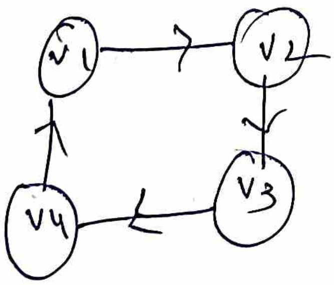
loop :- An edge where the starting and ending vertex are the same



V3 , V3
↓ starts ↓ ending

i.e have only one edge.

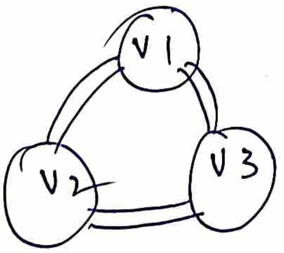
cycle :- A collection of edges



$\underline{v_1} \to v_2 \to v_3 \to v_4 \to \underline{v_1}$

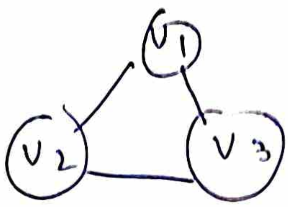
starting & ending vertex are same.

parallel edges :- If there is more the one edge between pair of edged

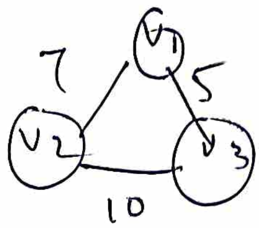


→ It is also multy graph.

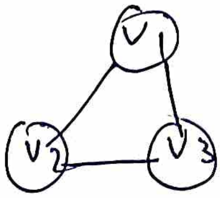
Simple graph :- If a graph does not contain any parallel edges then the graph is called simple graph



weighted graph :- If every edge contain some weight / cost then it is known as weighted graph

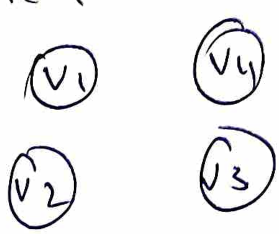


Isolated node :- An isolated node does not contain any edge cont node / vertex



v4 → v4 be the isolated node

Null graph / isolated graph :- All the vertex are isolated vertex



Order and size of a graph

Order of a Graph :- The no. of vertices in a graph is called as order of a graph

order of a graph is denoted by $|V|$

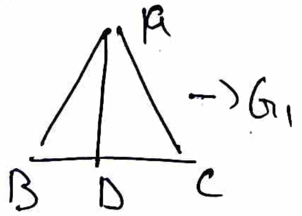
Size of a graph :- The no. of edges in a graph is called as size of a graph.

→ size of a Graph is denoted by $|E|$

Note :- ① A graph of order n and size m is called a (n, m) graph

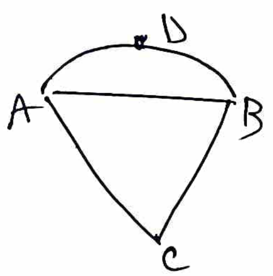
② $(4, 5)$ graph means it contains 4 vertices and 5 edges.

Eg :- ① Find the order and size of the follow graph

(a) 

order of graph $G_1 = 4$

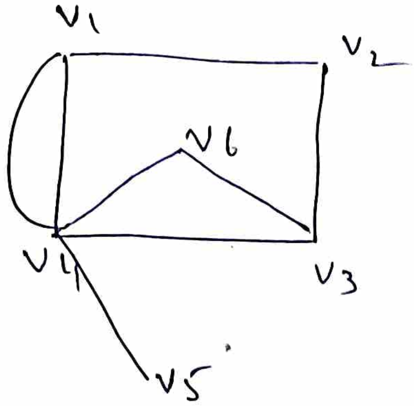
size " " $G_1 = 5$

(b) 

order = 4

size = 5

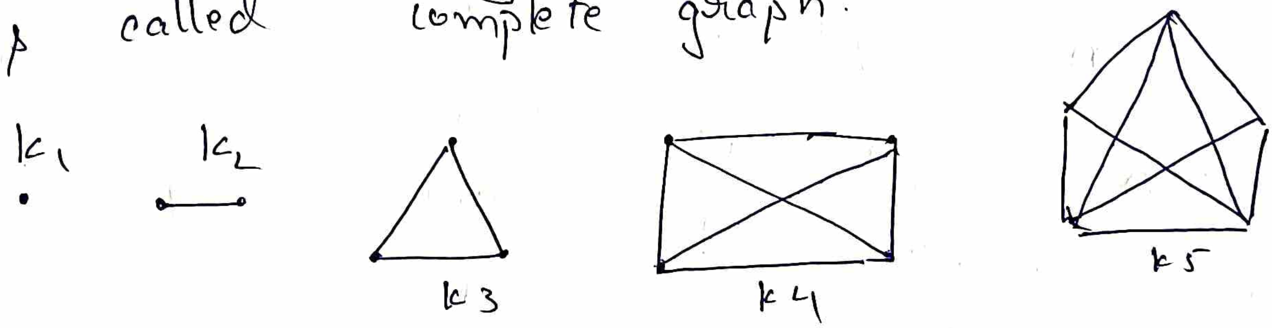
(c)



order = 6
size = 8

Types of graphs (simple graph)

(1) **complete graph** :- If each vertices is connected to every other vertex then the graph is called complete graph.



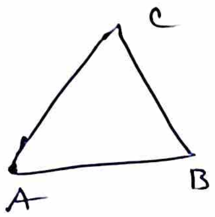
Note :- total no. of edges in complete graph is $\frac{n(n-1)}{2}$

Eg:- can complete graph with 8 vertices have 48 edges.

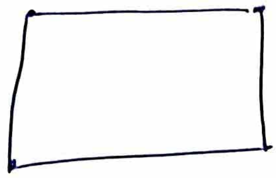
$$\frac{n(n-1)}{2} = \frac{8\times 7}{2} = \frac{56}{2} = 28$$

∴ have 28 edges only.

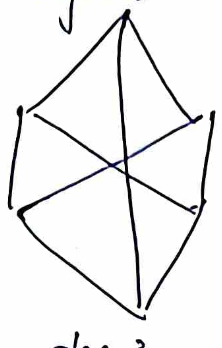
(2) <u>Regular graph</u> : A graph in which every vertex has same degree.



dog 2



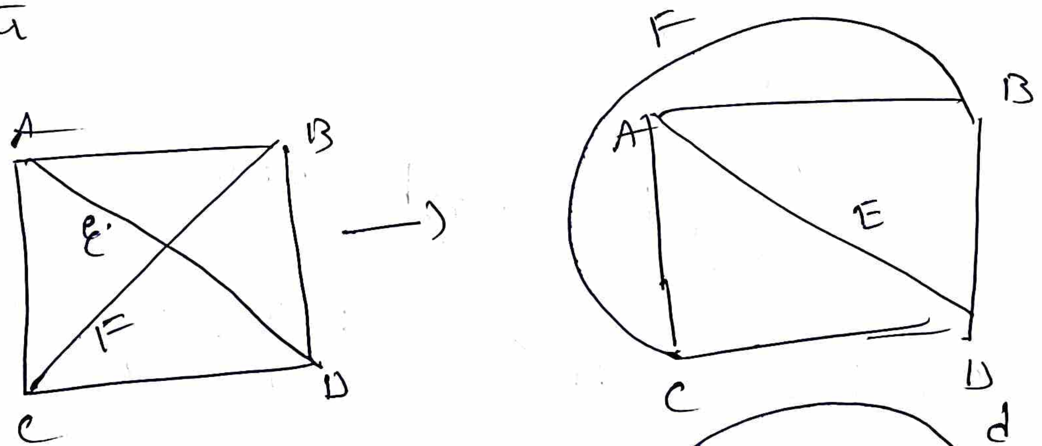
deg 2

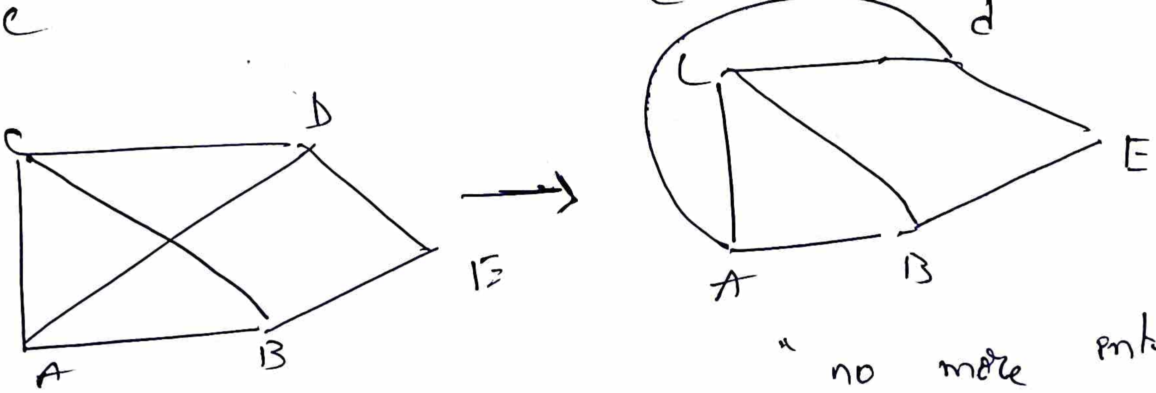


deg 3

Note :- Every complete graph is regular but not vice-versa.

(3) <u>planar graph</u> :- A graph is called to planar graph if it is possible to draw diagram of a given graph in such a way that no 2 edges intersect each other





" no more intersects

(5)

(4) **Bipartite Graph**: A graph is called bipartite if its vertex set V(G) can be partitioned into two non empty subsets V_1 & V_2 in such a way that each edge has its one end point in V_1 & other end point in V_2.

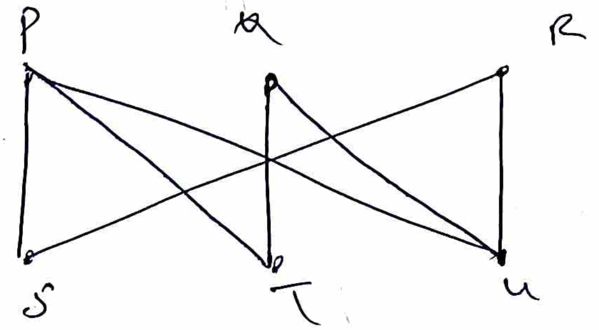
Eg. -

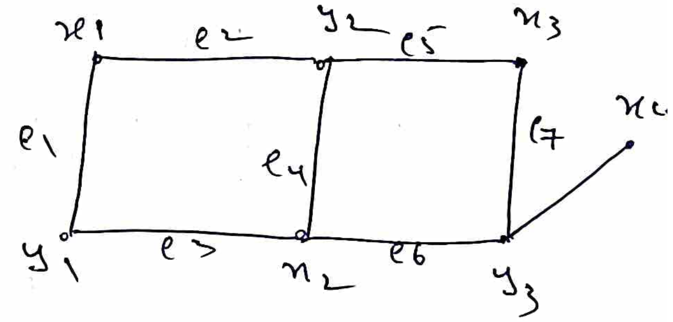
→ A Bipartite graph does not contain any self loops.

→ A Bipartite graph can be denoted by $K_{m,n}$.

where m contains the vertices in subset V_1
n " " " " V_2

Eg:- Draw $K_{2,3}$ and $K_{3,4}$ Bipartite graph
G(V,E)





$V = \{P, Q, R, S, T, U\}$

$V = \{x_1, y_1, x_2, y_2, x_3, y_3, x_4\}$

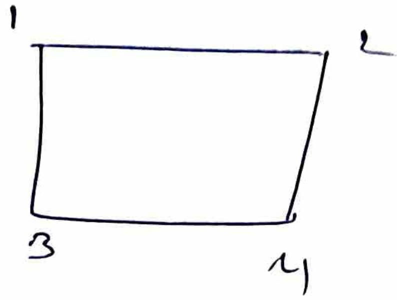
Eg $\{ V_1 = \{P, Q, R\}$
$V_2 = \{S, T, U\}$
not have common elements.

$V_1 = \{x_1, x_2, x_3, x_4\}$
$V_2 = \{y_1, y_2, y_3\}$.
not have common elements

(5) complete Bipartite graph:

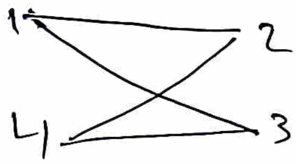
complete & Bipartitle both

$V = \{1, 2, 3, 4\}$



$V = \{1, 2, 3, 4\}$

$V_1 = \{1, 3\}$, $V_2 = \{2, 4\}$



(6) Null graph: A graph which contain only isolated nodes is called a null graph.

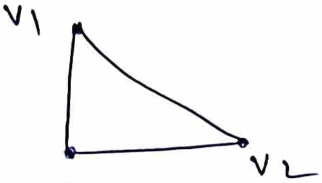
i.e The set of edge in a null graph is empty.

A null graph with only one vertex is called a trivial graph.

(6)

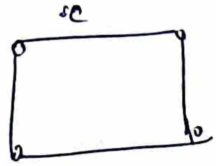
(7) **cycle graph** : The cycle C_n of length n, $n \geq 3$ consists of n vertices $V_1, V_2 -- V_n$ and edges $(V_1 V_2), (V_2, V_3) \dots (V_{n-1}, V_n)$ and (V_n, V_1)

(where ever we starting again come to that point only)



C3

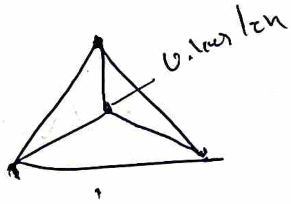
→ The cyclic graph with n vertices denoted as C_n.

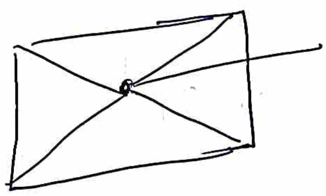


C4

wheel graph :- The wheel graph W_n (n>
is a graph formed by connecting a sin
universal vertex to all vertices of a cycle

(one vert
join all
remaining





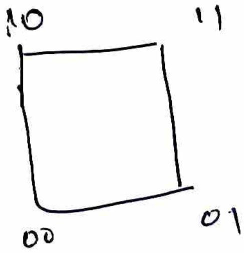
W..

n vertices & 2n-2 edges we ha

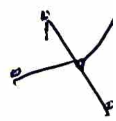
(8) **N-cube** : The N-cube is denoted by
vertices representing the 2^n bit strings
length n.

2^n vertices

$n 2^{n-1}$ edges

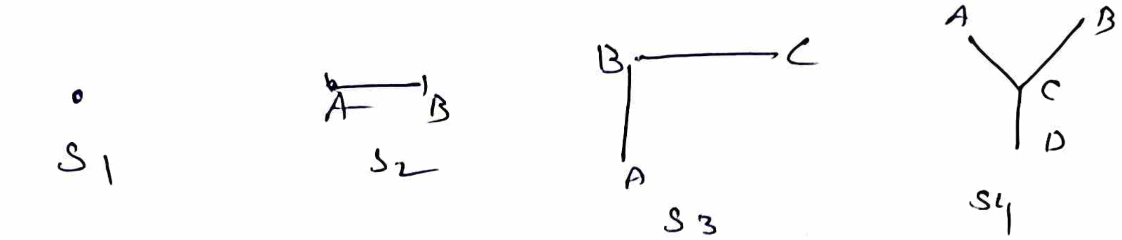


(9) **Star graph** : the star graph of ord
is a graph with (n-1) edges.
one vertex with degree n-1



→ A star graph with n vertices denoted as S_n

(1)



(10) <u>Finite Graph</u> & <u>Infinite graph</u>:

In a graph G, If the set of vertices and the set of edges an finite then that graph is called finite graph otherwise it is called an Infinite graph

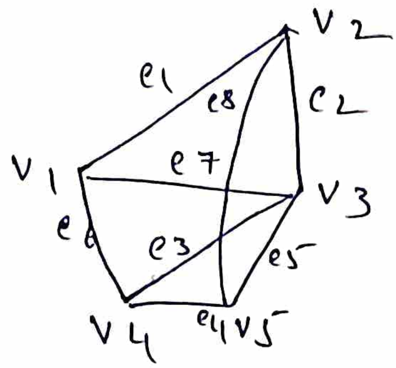
Sub graphs

Given two graphs $G(V,E)$ and $G_1(V_1,E_1)$. we say that G_1 is a sub-graph of G if the following conditions hold.

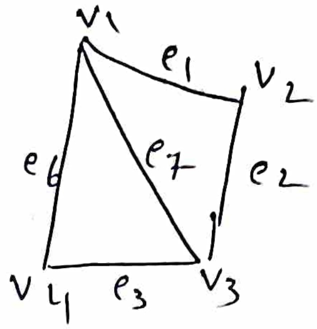
(i) All the vertices and all the edges of G_1 are in G i.e $V_1 \subseteq V$ & $E_1 \subseteq E$

(ii) each edge of G_1 has the same end vertices in G as in G_1

eg 1



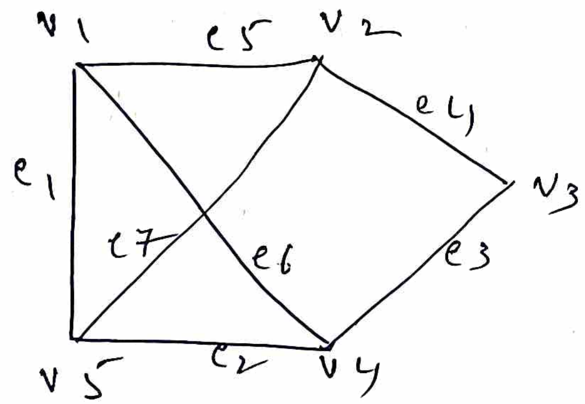
Graph - G(V, E)



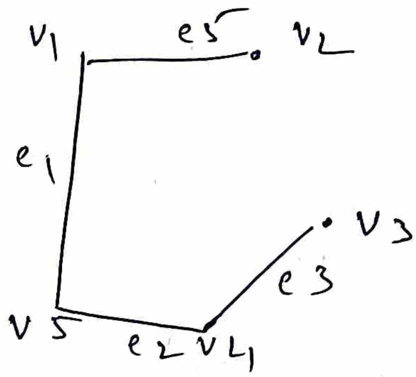
Graph $G_1(V_1, E_1)$

$G_1(V_1, E_1)$ is a subgraph of G(V, E)

2



Graph G(V, E)



Graph $G_1(V_1, E)$

$\therefore$ $G_1(V_1, E_1)$ is a subgraph of ~~$G(V_1, E_1)$~~ G(V, E)

Consequences from the definition of subgraph

(1) every graph is a subgraph of itself

(2) Every simple graph of n vertices is a subgraph of the complete graph K_n

(3) If G_1 is a subgraph of G_2 & G_2 is subgraph of G ~~theorem~~.

(4) A single vertex in a subgraph G is a ~~s~~ subgraph of G.

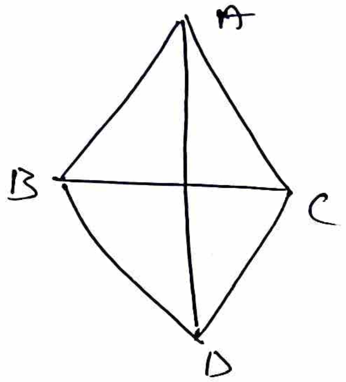
(5) A single edge in a graph G together with its end vertices is also a subgraph of G.

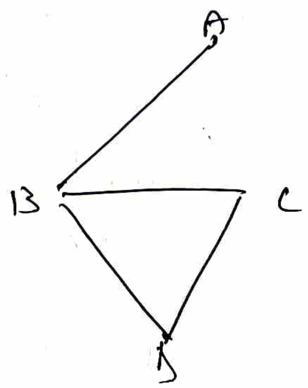
—.

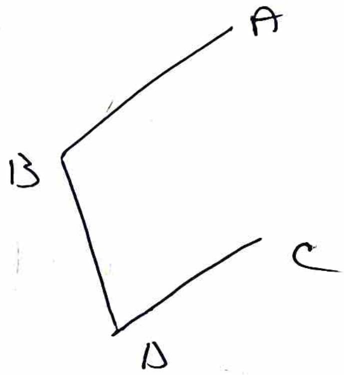
Spanning Subgraph

Given a graph $G = (V, E)$ If there is a subgraph $G_1 = (V_1, E_1)$ of G such that $V_1 = V$ then G_1 is called a spanning subgroup of G.

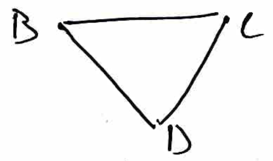
Eg :-



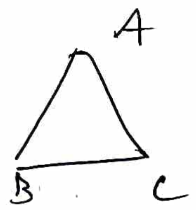




spanning sub graphs



↓
vertex - A
Missing



vertex - D
Missing

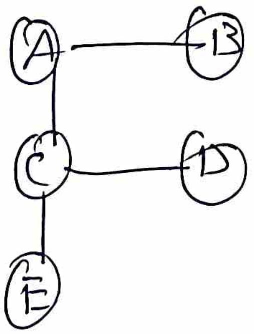
⟶ So not a spanni[ng] subgraph.

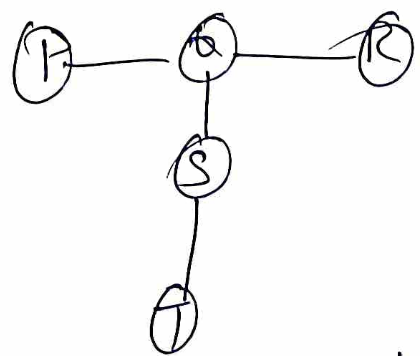
Isomorphic graphs

Two graphs are said to be isomorphic they satisfies the following conditions.

1. Both graphs must contain same no. of vertices
2. " " " " " edges
3. " " " " " deg sequence
4. one-to-one correspondence between the vertices of 2 graphs must be same. i.e for every vertex in graph 1 there should be an equivalent vertex in graph 2
5. each edge in graph 1 is equivalent to an edge in graph 2
6. Adjacent matrix of both graphs must be same.

eg:- ① check whether following graphs are isomorphic or not





① No. of vertices = 5
② edges = 4

① No. of vertices = 5
② edges = 4

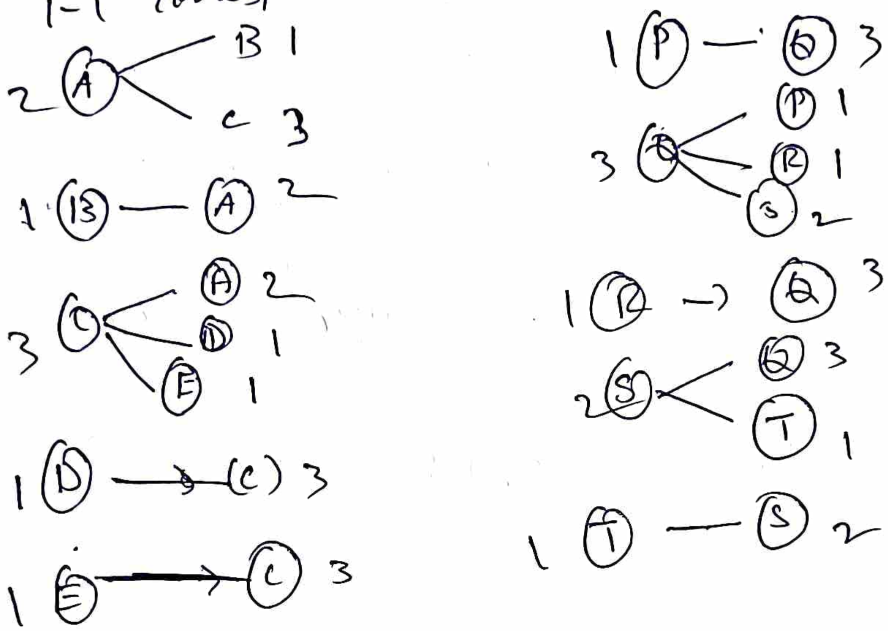
(3) Degree sequence

$(A, B, C, D, E) = (2, 1, 3, 1, 1) \rightarrow (3, 2, 1, 1, 1)$ decreasing order

$(P, Q, R, S, T) = (1, 3, 1, 2, 1) \rightarrow (3, 2, 1, 1, 1)$

1-1 correspondence   1-1 correspondence

(4)



C ⟷ Q

A ⟷ S

B ⟷ T   ∴ 1-1 correspondence is ens

D ⟷ P

E ⟷ R

(5) edge preserving should be satisfied

A — B ⟷ S — T

A — C ⟷ S — Q

C — D ⟷ Q — P

C — E ⟷ Q — R

satisfied

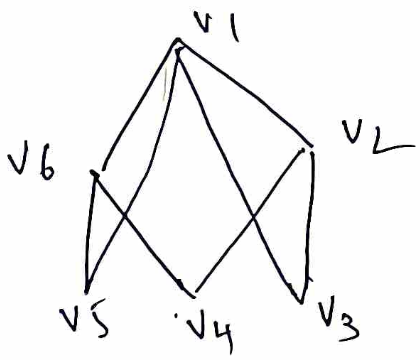
(6) Adjecancy matrix

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 1 & 1 & 0 & 0 \\ B & 1 & 0 & 0 & 0 & 0 \\ C & 1 & 0 & 0 & 1 & 1 \\ D & 0 & 0 & 1 & 0 & 0 \\ E & 0 & 0 & 1 & 0 & 0 \end{array} \qquad \begin{array}{c|ccccc} & S & T & Q & P & R \\ \hline S & 0 & 1 & 1 & 0 & 0 \\ T & 1 & 0 & 0 & 0 & 0 \\ Q & 1 & 0 & 0 & 1 & 1 \\ P & 0 & 0 & 1 & 0 & 0 \\ R & 0 & 0 & 1 & 0 & 0 \end{array}$$

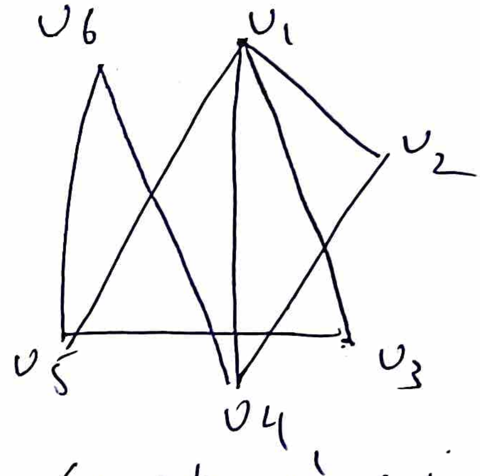
adjcent matrices are also same

∴ These two graphs are isomorphism

(2)



Graph G



Graph G' (V', E')

verify the above graphs G and G' are isomorphic or not.

sol :- (i) Number of vertices in |V| = 6

(ii) No. of vertices in |V'| = 6

(2) No. of edges $|E| = 8$

(2) No. of edges $|E'| = 8$

(3) Degree of all vertices in G is $\{v_1, v_2, v_3, v_4, v_5, v_6\}$

$= \{4, 3, 2, 2, 2, 3\}$ (v_1, v_2, v_3, v_4, v_5, v_6)

$= \{4, 3, 3, 2, 2, 2\}$

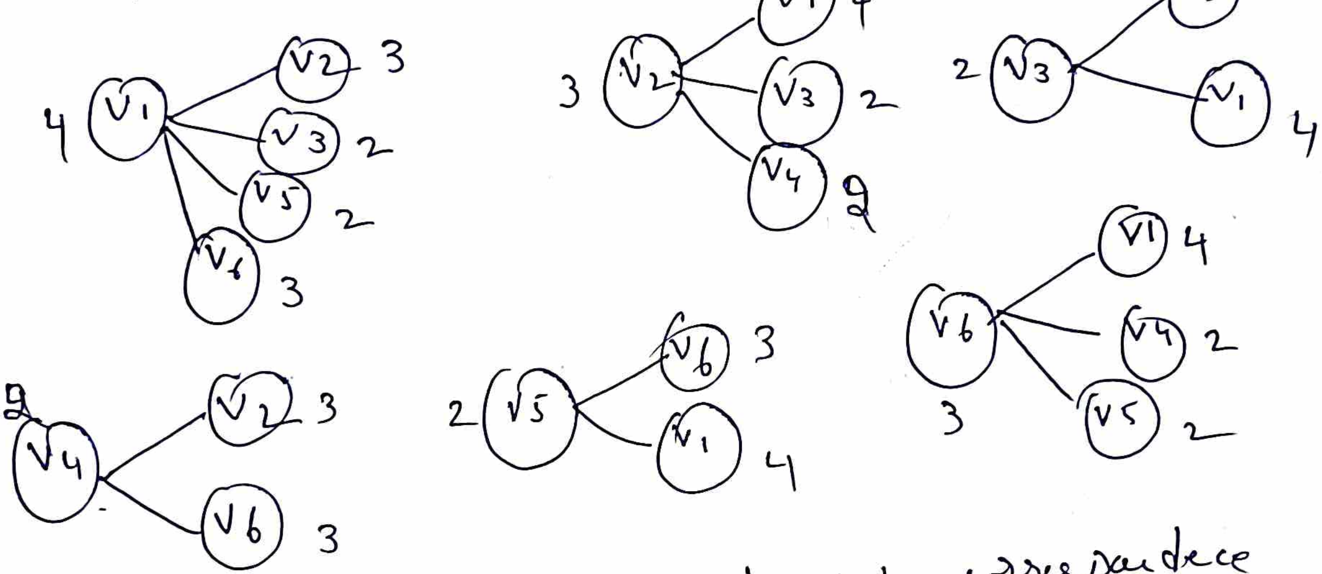
(3) deg of all vertices in graph $G'(u_1, u_2, u_3, u_4, u_5, u_6)$

$= \{4, 2, 2, 3, \cancel{3}, 2\}$

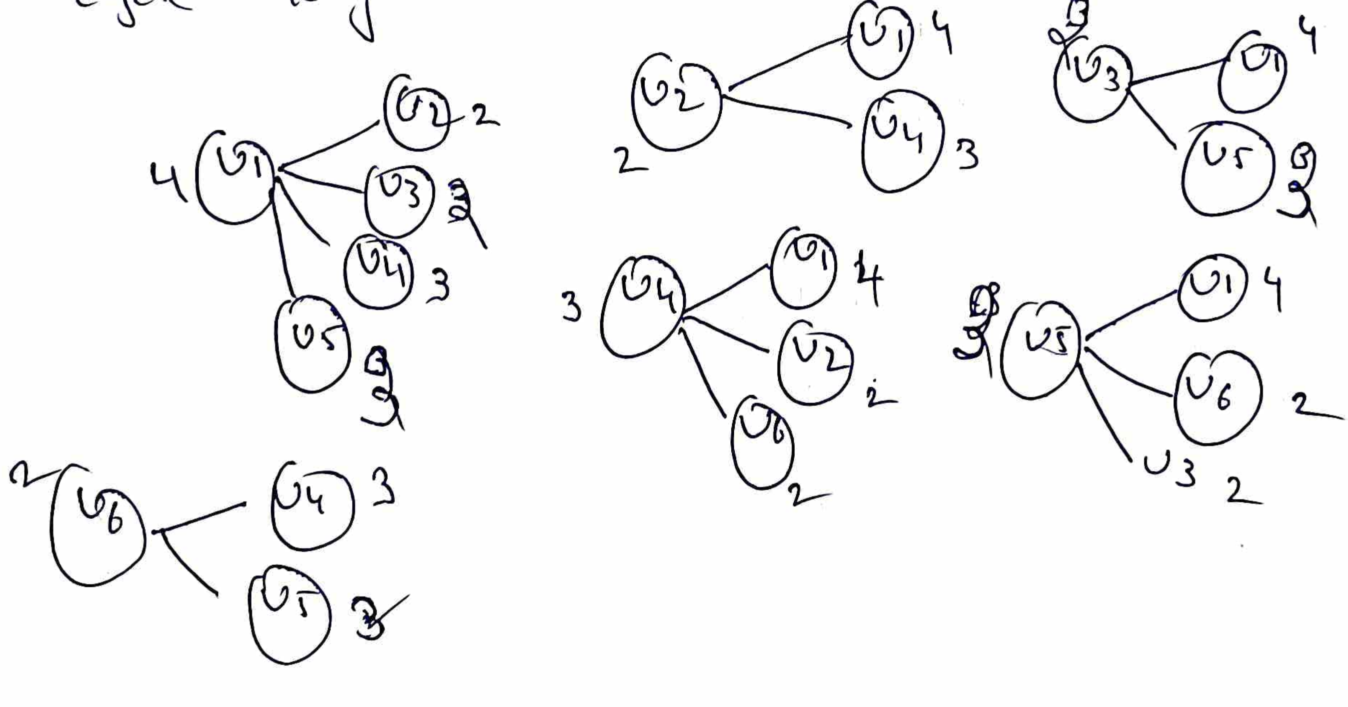
$\{4, 3, 3, 2, 2, 2\}$

(4) ~~cycle~~ length ~~in~~ ~~G~~ 1-1 correspondence in G

~~$v_4 - v_2 - v_5$~~



~~cycle~~ ~~length~~ ~~in~~ ~~G~~ & 1-1 correspondence in G'



$v_1 \longleftrightarrow u_1$

$v_2 \longleftrightarrow u_4$

$v_3 \longleftrightarrow u_2$

$v_4 \longleftrightarrow u_6$

$v_5 \longleftrightarrow u_3$

$v_6 \longleftrightarrow u_5$

$\therefore$ one-to-one correspondence exist.

(5) edge preserving

vertex preserving

$\{v_1, v_2, v_3, v_4, v_5, v_6\}$

vertex preserving

$\{u_1, u_4, u_2, u_6, u_3, u_5\}$

(6) edge preserving

edge preserving

$\{v_1, v_2\} \longleftrightarrow \{u_1, u_4\}$

$\{v_1, v_3\} \longleftrightarrow \{u_1, u_2\}$

$\{v_1, v_5\} \longleftrightarrow \{u_1, u_3\}$

$\{v_1, v_6\} \longleftrightarrow \{u_1, u_5\}$

$\{v_2, v_3\} \longleftrightarrow \{u_4, u_2\}$

$\{v_4, v_2\} \longleftrightarrow \{u_6, u_4\}$

$\{v_4, v_6\} \longleftrightarrow \{u_6, u_5\}$

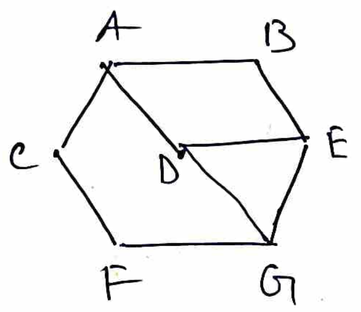
$\{v_5, v_6\} \longleftrightarrow \{u_3, u_5\}$

Adjacency matrix

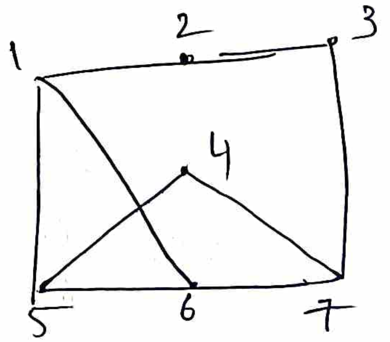
$$
\begin{array}{c|c}
 & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 &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

$\therefore$ G and G' are Isomorphic to each other

(3)



Graph - G_1



Graph - G_2

(1) No. of vertices : 7

(2) No. of edges : 9

(3) Degree of G_1

{ A, B, C, D, E, F, G }

{ 3, 2, 2, 3, 3, 2, 3 }

(1) No. of vertices = 7

(2) No. of edges = 9

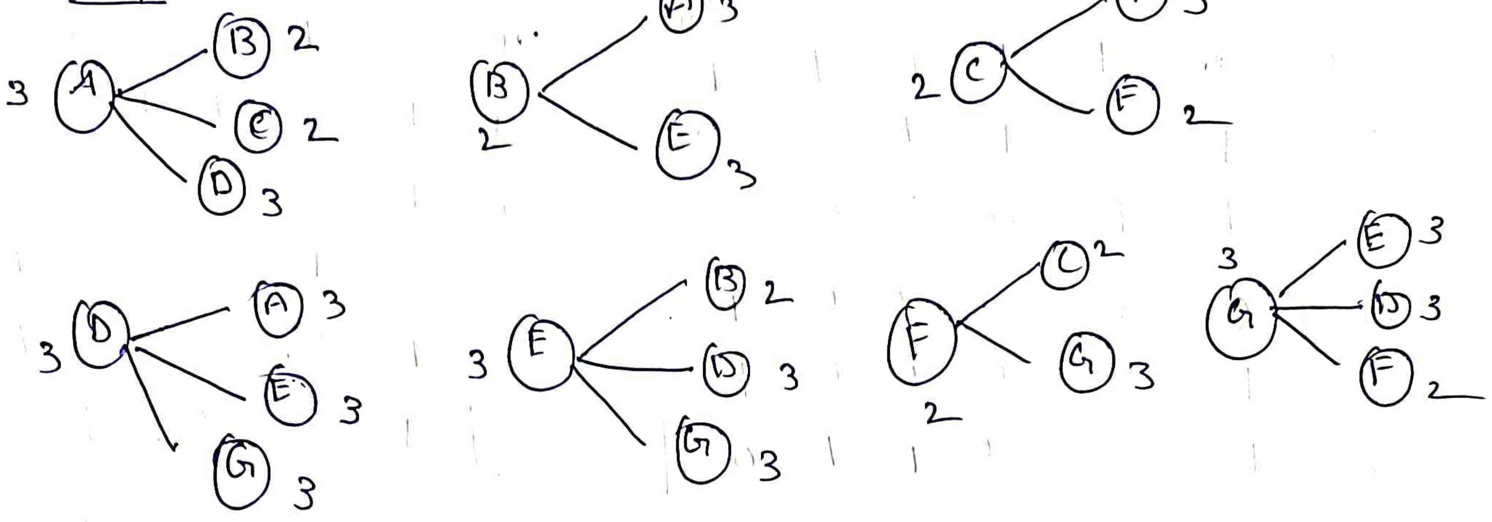
(3) degree of G_2

{ 1, 2, 3, 4, 5, 6, 7 }

{ 3, 2, 2, 2, 3, 3, 3 }

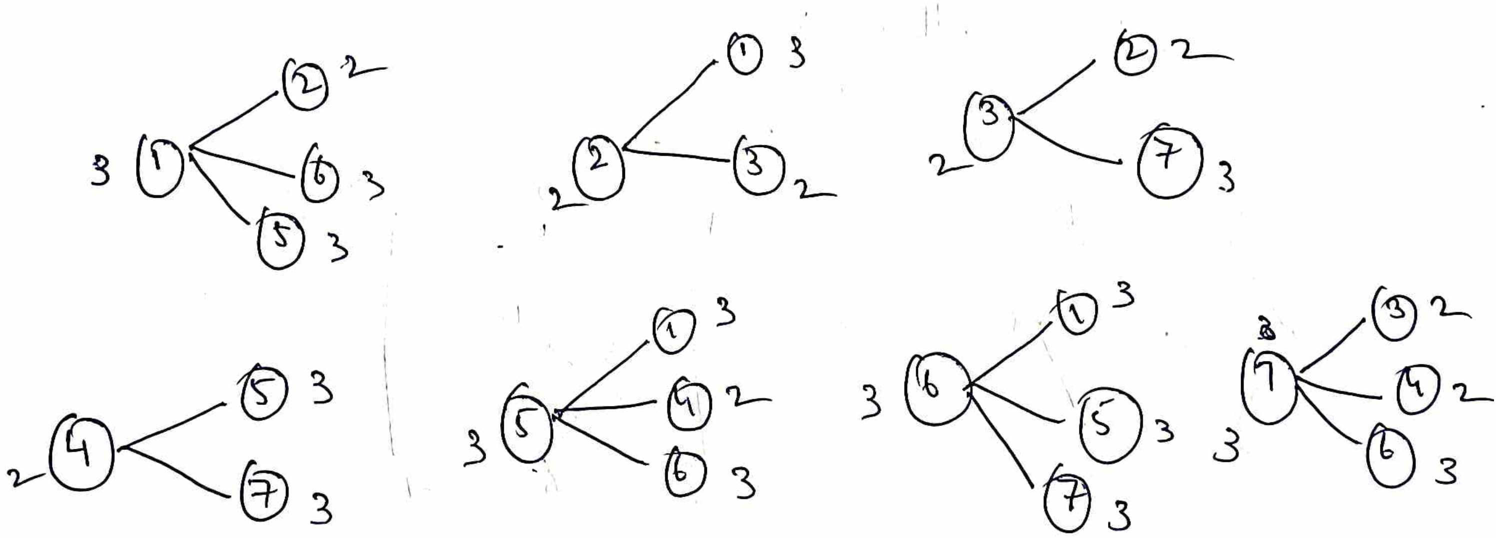
(4) 1-to-1 correspondence of G_1

G_1



G_2

one-to-one correspondence of G_2



A ⟷ 7

B ⟷ 4

C ⟷ 3

D ⟷ 6

E ⟷ 5

F ⟷ 2

G ⟷ 1

(8) vertex preserving G_1 (8) vertex preserving

$\{A, B, C, D, E, F, G\}$ $\{7, 4, 3, 6, 5, 2, 1\}$

(6) edge preserving

$\{A, B\} \longleftrightarrow \{7, 4\}$

$\{A, C\} \longleftrightarrow \{7, 3\}$

$\{A, D\} \longleftrightarrow \{7, 6\}$

$\{B, E\} \longleftrightarrow \{4, 5\}$

$\{E, G\} \longleftrightarrow \{5, 1\}$

$\{D, G\} \longleftrightarrow \{6, 1\}$

$\{D, E\} \longleftrightarrow \{6, 5\}$

$\{F, G\} \longleftrightarrow \{2, 1\}$

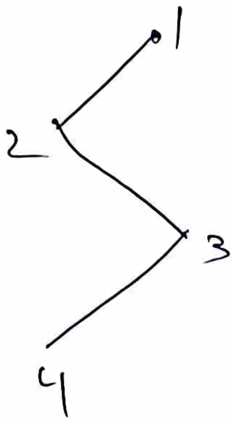
$\{C, F\} \longleftrightarrow \{3, 2\}$

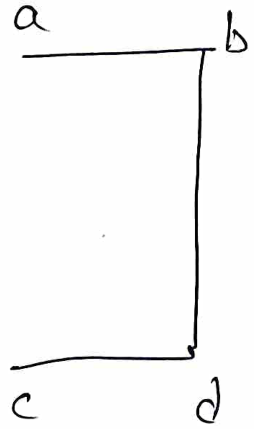
(7) Adjacency matrix

$$\begin{array}{c|c} & \begin{array}{ccccccc} A & B & C & D & E & F & G \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array} \quad \begin{array}{c|c} & \begin{array}{ccccccc} 7 & 4 & 3 & 6 & 5 & 2 & 1 \end{array} \\ \begin{array}{c} 7 \\ 4 \\ 3 \\ 6 \\ 5 \\ 2 \\ 1 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$\therefore$ G_1 & G_2 are Isomorphic

→ Show that the given pair of graphs are isomorphic





G_1 G_2

① No. of vertex : 4

② No. of edges 3

③ degree of all vertex

{1, 2, 3, 4}

{1, 2, 2, 1}

① No. of vertex = 4

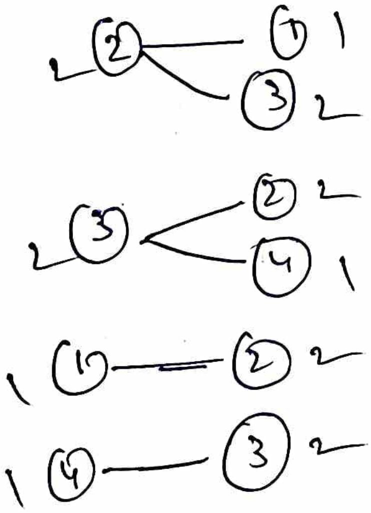
② No. of edges = 3

③ {a, b, c, d}

{1, 2, 2, 1}

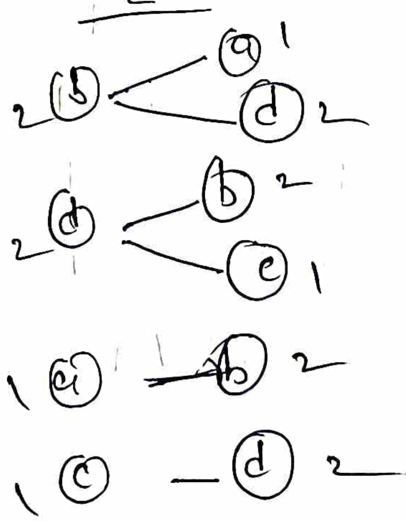
④ one-to-one correspondence

G_1



④ 1-1 correspondence

G_2



$2 \longleftrightarrow b$

$1 \longleftrightarrow a$

$3 \longleftrightarrow d$

$4 \longleftrightarrow c$

(5) vertex preserving G_1

$\{2, 1, 3\ 4\}$

(5) vertex preserving G_2

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

(6) edge preserving

$\{1,2\} \longleftrightarrow \{a,b\}$
$\{2,3\} \longleftrightarrow \{b,d\}$
$\{3,4\} \longleftrightarrow \{d,c\}$

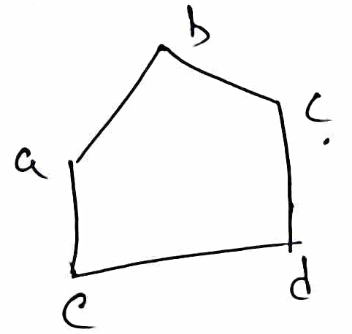
edge preserving satisfies.

(7) Adjacency matrix G_1 & G_2

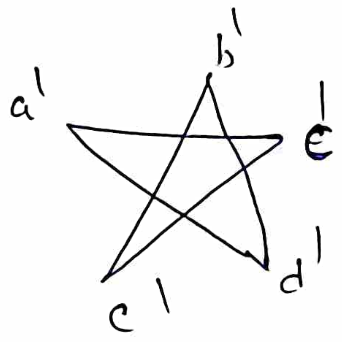
$$\begin{array}{c|c} & \begin{matrix} 2 & 1 & 3 & 4 \end{matrix} \\ \begin{matrix} 2 \\ 1 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \begin{matrix} b & a & d & c \end{matrix} \\ \begin{matrix} b \\ a \\ d \\ c \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$\therefore$ G_1 & G_2 are isomorphic

→ show that the given pair of graphs are isomorphic



G_1



G_2

Sol:- ① No. of vertices = 5 ① No. of vertices = 5

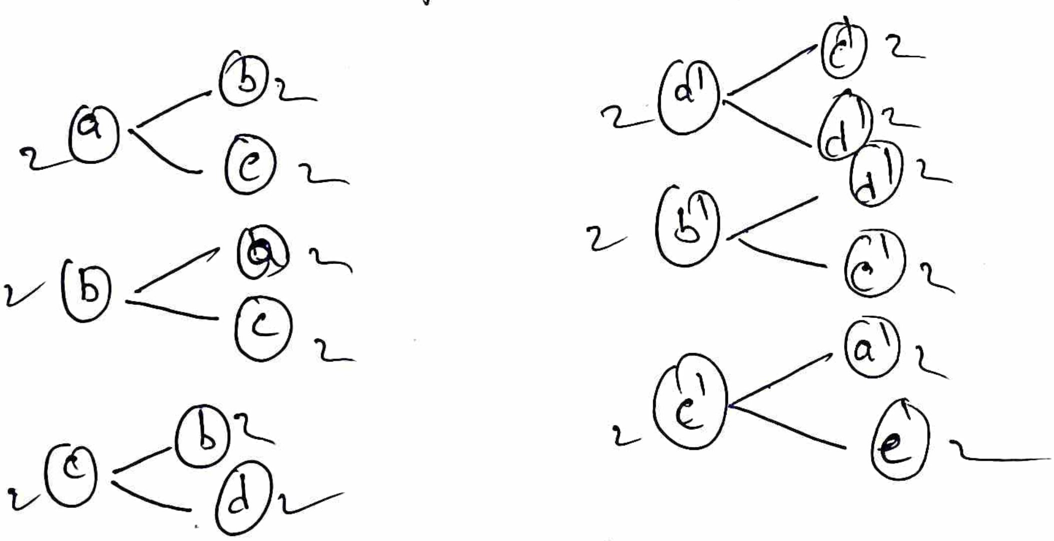
② No. of edges = 5 ② No. of edges = 5

③ Degrees of G_1 ③ deg of G_2

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

$\{2, 2, 2, 2, 2\}$ $\{2, 2, 2, 2, 2\}$

④ 1-1 correspondence of G_1 & G_2



$a \longleftrightarrow b'$

$b \longleftrightarrow d'$

$e \longleftrightarrow a'$

$d \longleftrightarrow c'$

$e \longleftrightarrow e'$

8 Adjacency matrix of G_1 & G_2

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} &
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\qquad
\begin{array}{c|c}
 & \begin{array}{ccccc} b' & d' & a' & c' & e' \end{array} \\
\begin{array}{c} b' \\ d' \\ a' \\ c' \\ e' \end{array} &
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

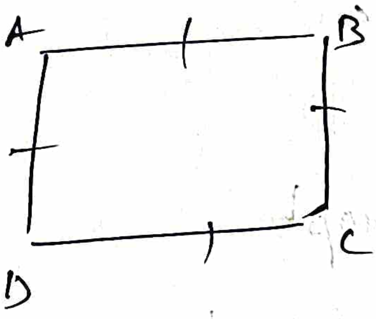
$\therefore$ G_1 & G_2 are isomorphic

Eulerian graph

Euler graph :- A graph is said to be euler graph if it has closed (Euler circuit) on it.

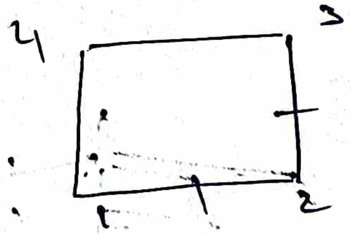
Euler path (starting ending vertex must be same) & **closed euler path** : In closed Euler path starting and ending vertex must be same.

Eg :-



→ All edges visited must once

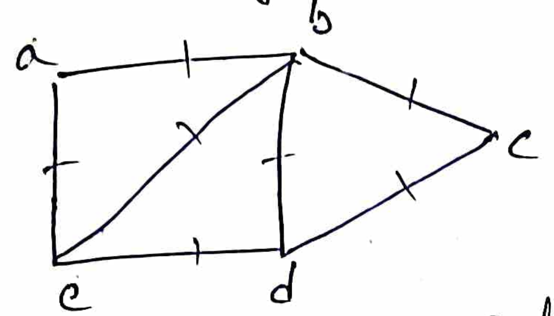
Euler path :- If it includes all edges once and vertex may be repeated.



Note :- ① If a graph is euler graph all the vertex must have even degree

② If there exactly 2 vertex of odd degree no euler graph but Euler path that will start from one vertex of odd degree end with other vertex of odd degree.

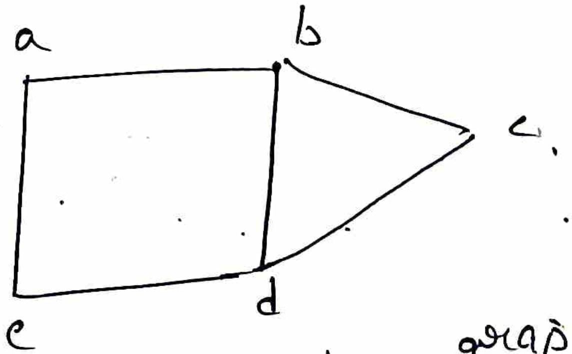
Eg:- ①



It is not euler graph because degree not have even degree.

e – a – b – e – d – b – c – d

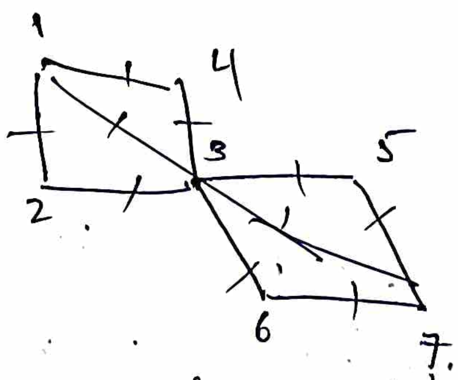
②



Not a Euler graph

d – e – a – b – c – d – b

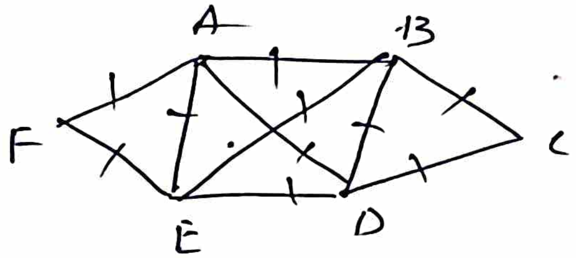
③



not Euler graph

1 – 4 – 3 – 2 – 1 – 3 – 5 – 7 – 6 – 3 – 7

3



Euler graph.

A - B - D - C - B - E - D - A - E - F - A.

Hamiltonian graph

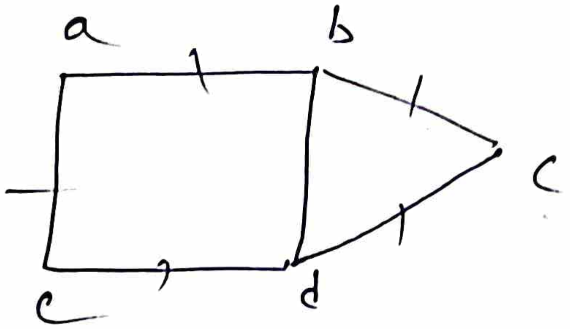
Def :- A graph ~~is~~ said to be Hamiltonian if there exist closed Hamiltonian path or (Hamiltonian circuit)

<u>closed Hamiltonian path</u> : If starting and ending vertex same.

<u>Hamiltonian path</u> :- Each and every edge must be visited once (except starting vertex)

eg:-

①



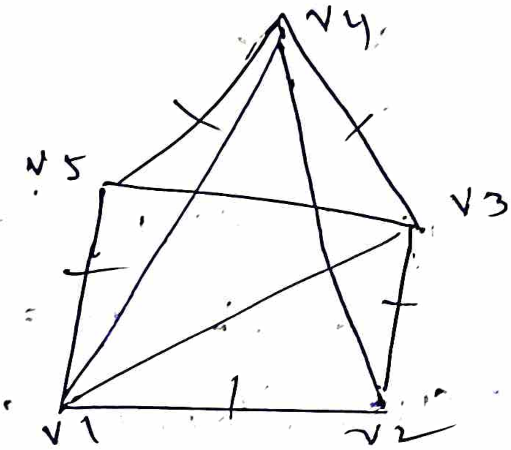
a - b - c - d - e - a

Hamiltonian graph.

a - b - c - d - e {H.P}

a - b - c - d - e - a {closed H.T path}

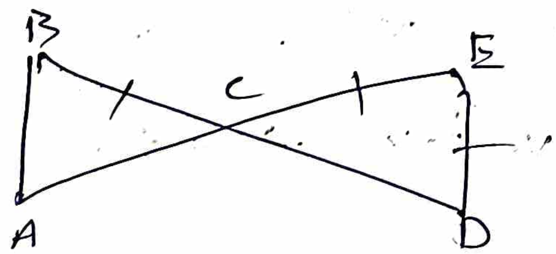
②



V1 - V2 - V3 - V4 - V5 - V1

closed hamiltonian path

③

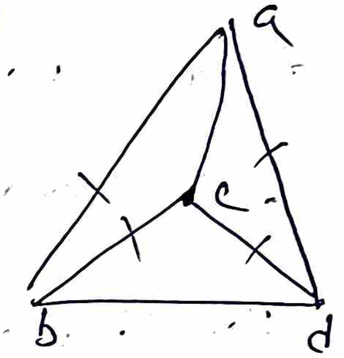


A-B - C - E - D

Hamiltonian path

not hamiltonian graph

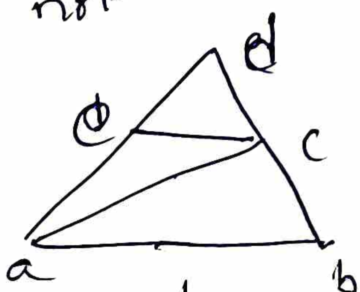
④



a - b - c - d - a

H.G

④

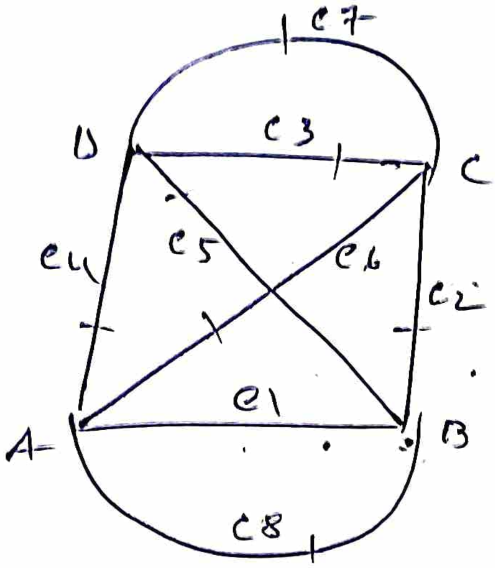


a - b - c - e - d

not H.G

Give an example of following

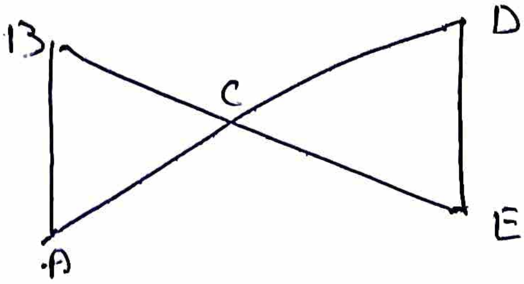
(1) Eulerian & Hamiltonian



closed Euler path $A e_1 B e_2 C e_3 D e_4 A e_6 C e_7 D e_5 B e_8 A$

closed hamiltonian $A e_1 B e_2 C e_3 D e_4 A$

(2) Eulerian but not Hamiltonian

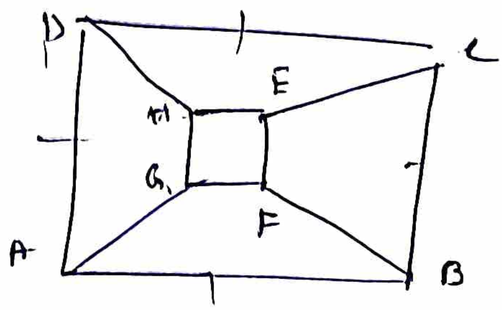


E.C.P A - B - C - E - D - C - A

H.P A - B - C - D - E

not closed.

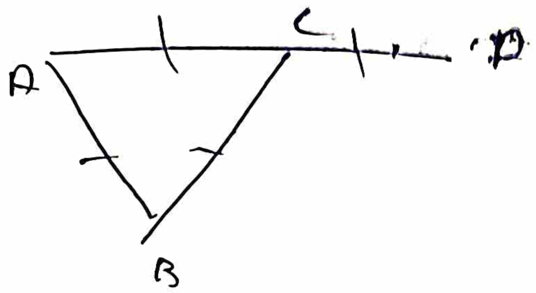
(3) Hamiltonian but not Euler:



CH.P A - B - C -> D - H - E - F -> G - A

all vertex are visited starting & ending vertex same not Euler

(4) Neither Hamiltonian & Eulerian graph



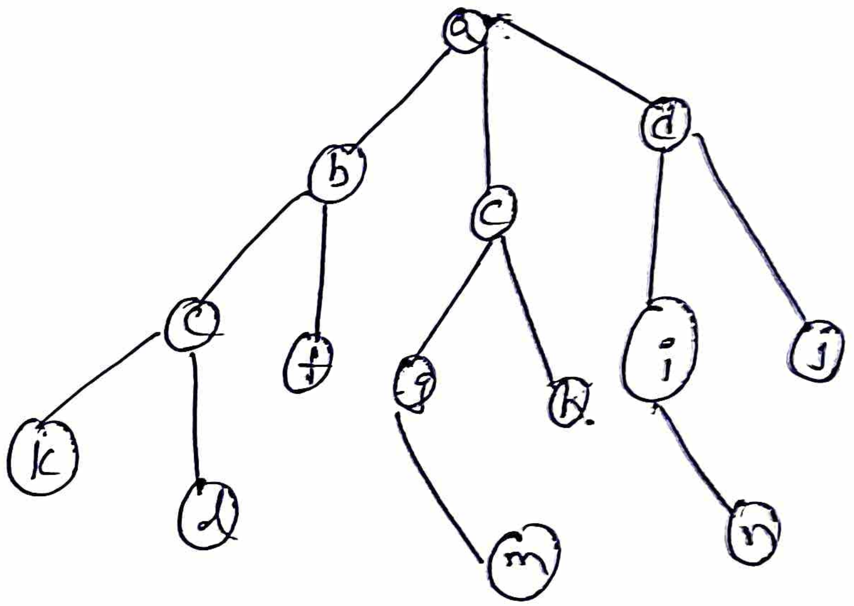
No Euler graph

D – C – A – B – C   E.P

H.P   A – B – C – D   H.P not closed path

TREES

A tree is connected a cyclic undirected graph. A graph with N number of vertices has (n-1) edges



properties of Tree

① Tree with n vertices has (n-1) edges

② A graph is a tree if it is minimal -ly connected.

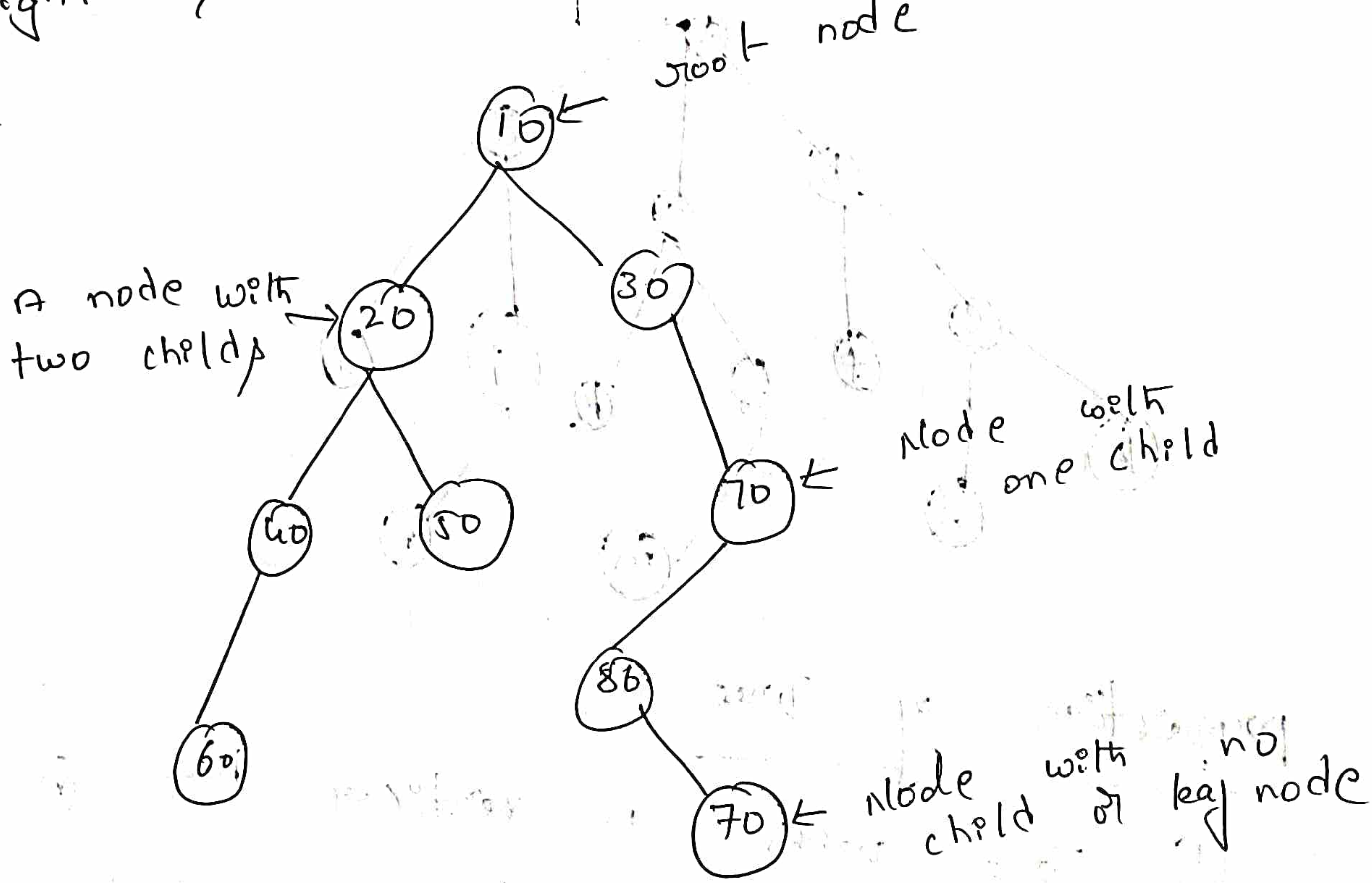
③ every tree is graph but every graph is not a tree

④ There is only one path between each pair of vertices of a tree.

Binary tree and Types of Binary trees

Binary tree :- A Binary tree is a finite set of nodes. It is either empty or consists of a root and two disjoint binary trees called the left sub tree and right sub tree

Eg:-



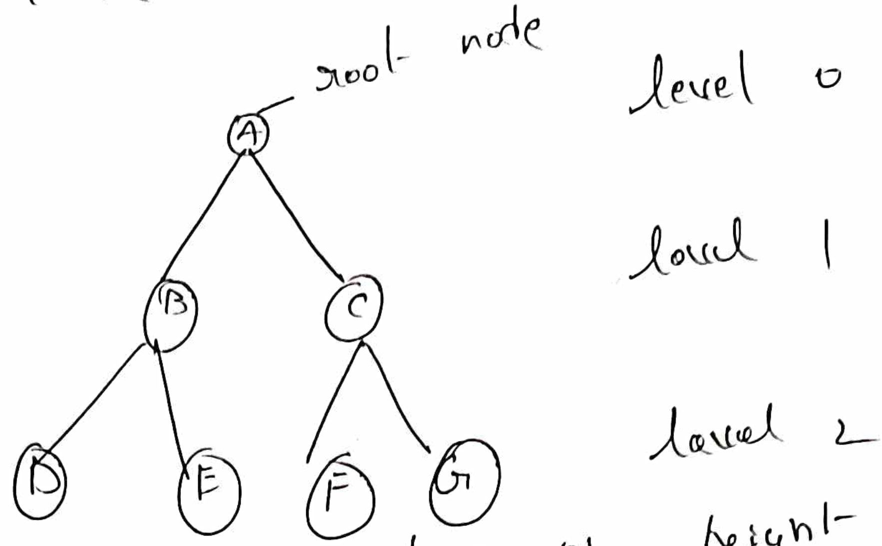
Binary tree diagram

Types of Binary trees:

(1) Full Binary tree :- A Binary tree is a full Binary tree if and if of each non-leaf node has exactly (a) two children

(iii) All leaf nodes are at the same level

Ex :-

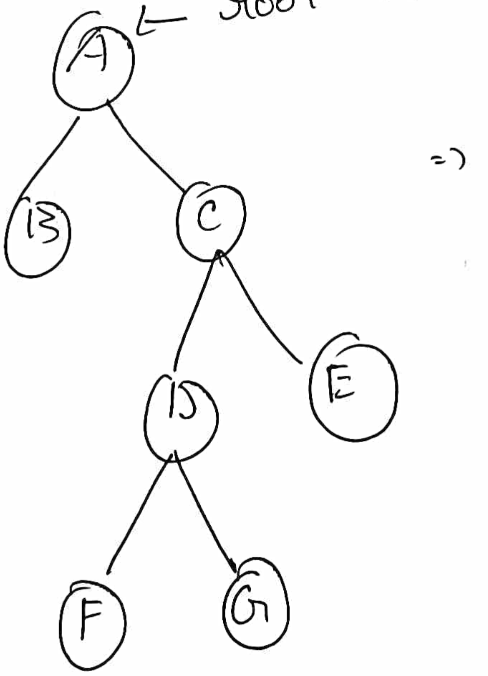


A full Binary tree of height '2' fig

(2) **strictly Binary tree :-** strictly Binary tree is a binary tree if and only if tree is a binary tree if every non-leaf node in the binary tree ~~is~~ and filled with left ~~sub-trees~~ sub-trees and right sub-trees.

In strictly Binary tree all are leaf nodes are not at the same level

Ex :-



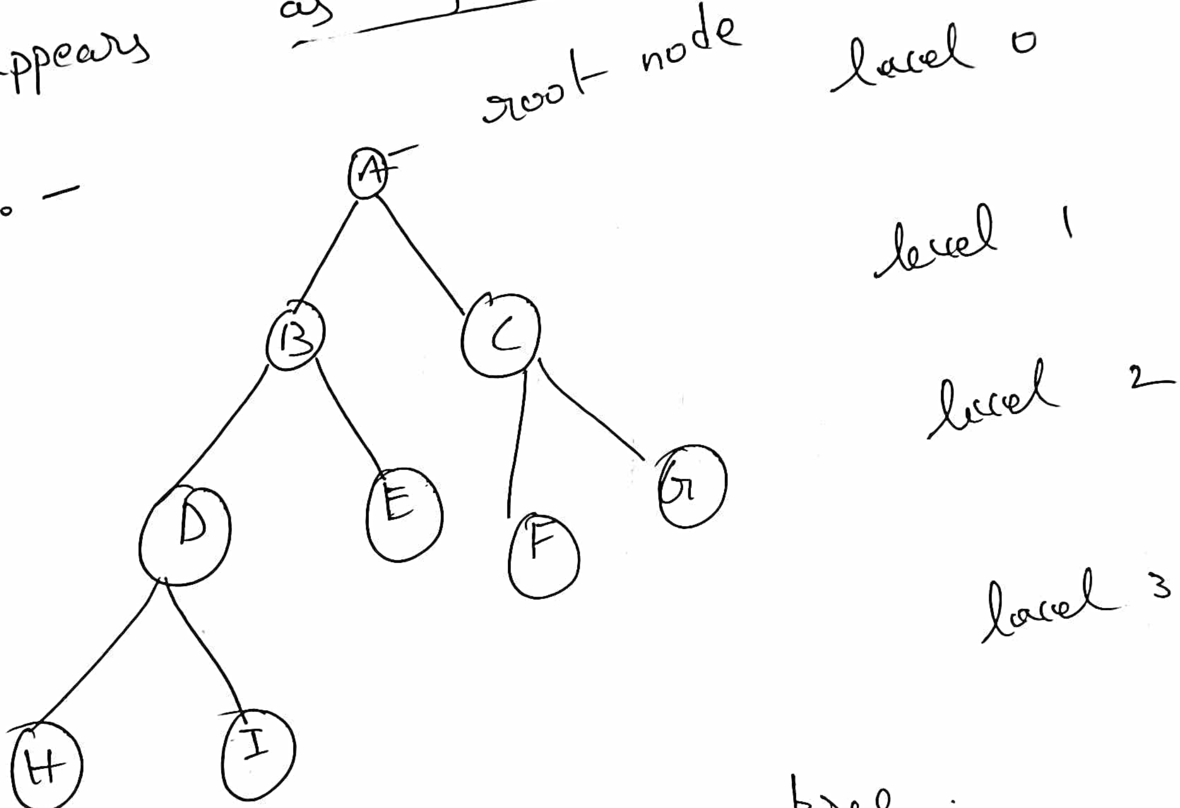
=> B, F, G, E are leaf nodes

=> Here A, C, D are non-leaf nodes each have two childs.

(3) <u>complete binary tree</u>: A complete binary tree is a binary tree if and only if (i) All the levels in complete binary tree except possibly the last level have maximum no. of nodes

(ii) All the nodes at the least level appears as far left as possible.

Eg:-



complete Binary tree.

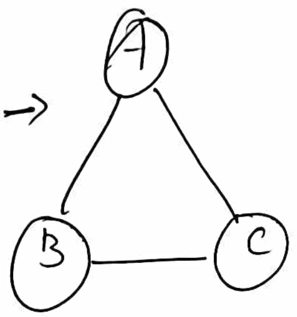
(4) <u>extended Binary tree</u>

Spanning trees

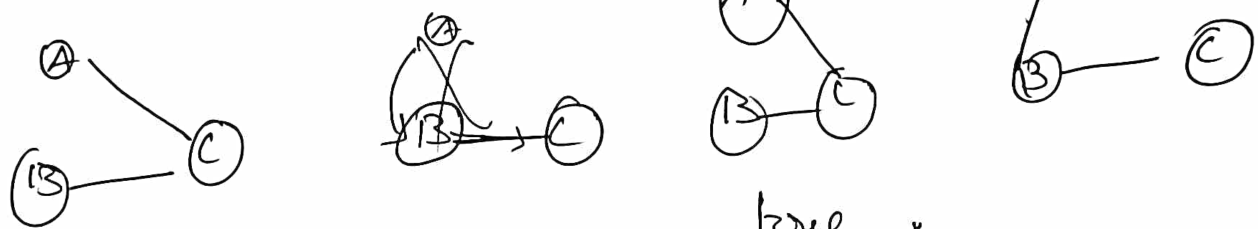
Def :- Spanning tree is a subset of graph G which has all vertices covered with min possible no. of edges

Hence A spanning tree does not have cycle and it cannot be disconnected

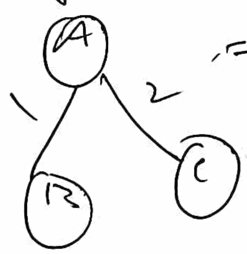
Eg :- G-graph →



⇒ spanning tree :-



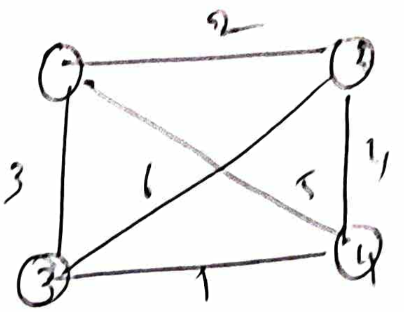
weight of spanning tree : weight of spanning tree w(T) is the sum of weights of all edges in T



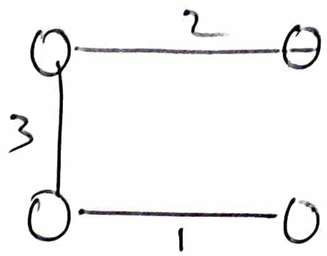
$= 1 + 2$

$= 3$

⇒ Minimum spanning tree (MST) is a spanning tree with smallest possible weight



M.S Trees from weighted graph :



weight = 1+2+3
= 6

Eg :-

① M.S.T can be use in design of network {least cost path} airline ~~routes~~ routes. the vert
② " finding cities and edges – route between cities. and
- cities represents edges – Route between cities

=> MST implemented with no cycles.

Minimum spanning trees can be constructed using

① kruskal's Algorithm. (uses edges)
② prim's algorithm. (uses ~~edges~~ vertex)

Imp. Representation of Graphs

There are 3 types of Representation graphs

1. Adjacency Matrix Representation
2. Incidence Matrix Representation
3. Incidence Matrix Representation

(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 $A = (a_{ij})_{n \times n}$ of G is an $n \times n$ symmetric matrix defined by the following elements.

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

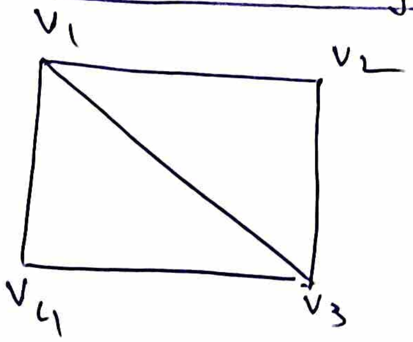
(or)

$$a_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are connected by edge of an } G \\ 0, & \text{otherwise} \end{cases}$$

It is denoted by A_G (or) $A(G)$

Eg:- If G is a graph then its adjacency matrix A_G (or) $A(G)$ is

(a) Undirected graph :



It is undirected graph because each and every edge in the graph not-directed. so $G(V, E)$ is undirected graph.

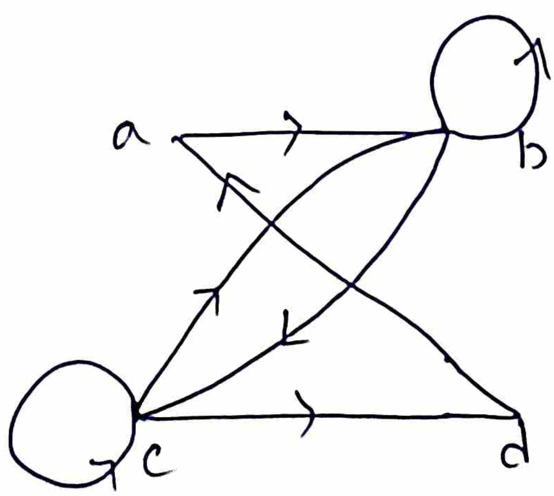
Here G be the graph
V be the vertices
E be the edges

adjacency matrix is

$$A(G) = a_{ij} \quad \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 0 & 1 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 1 & 0 & 1 \\ v_4 & 1 & 0 & 1 & 0 \end{array}$$

symmetric property exists.

(b) Directed graph :-



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

(2) Incidence Matrix Representation :

Let $G(V, E)$ is an undirected graph with n vertices ordered from $v_1, v_2 \dots v_n$ ~~em~~ and m edges ordered from $e_1, e_2 \dots e_m$ then the Incidence Matrix is Represented of order $n \times m$ is denoted by I_G or $I(G)$. The elements of the matrix

$$I_G = [M_{ij}]_{n \times m} \text{ is defined as}$$

$$M_{ij} = \begin{cases} 1, & \text{when edge } e_j \text{ is incident on } v_i \\ 0, & \text{otherwise} \end{cases}$$

The following basic properties of an Incidence matrix are :-

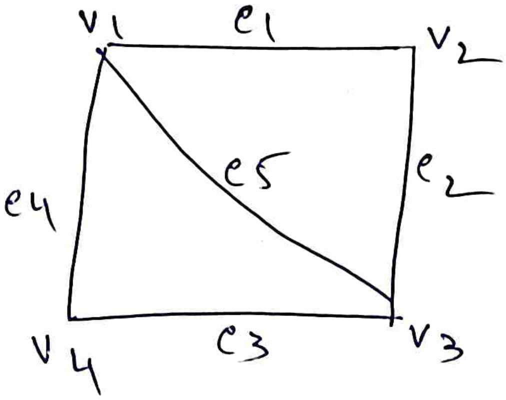
(1) Each column of incidence matrix contains exactly two 1's

(2) A row with all 0's represents an isolated vertex.

(3) A row with single one represents an pendant vertex.

(4) The total no. of 1's in ith row represents the degree of the ith vertex.

Eg :- (1)



G(V, E)

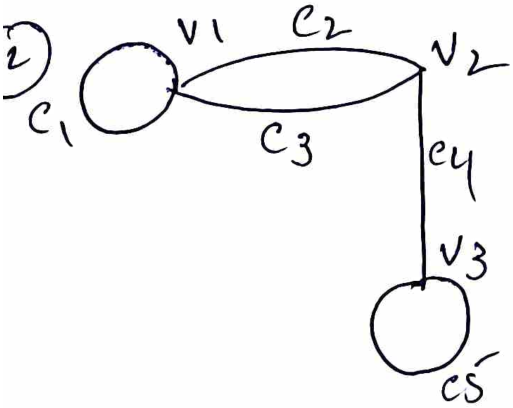
I_G (or) $I(G) = [M_{ij}]_{n \times m}$

Here n be the no. of vertices
m be the no. of edges.

$$M_{ij} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

(e_1 incident on v_1 & v_2)

(2)



$$M_{ij} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

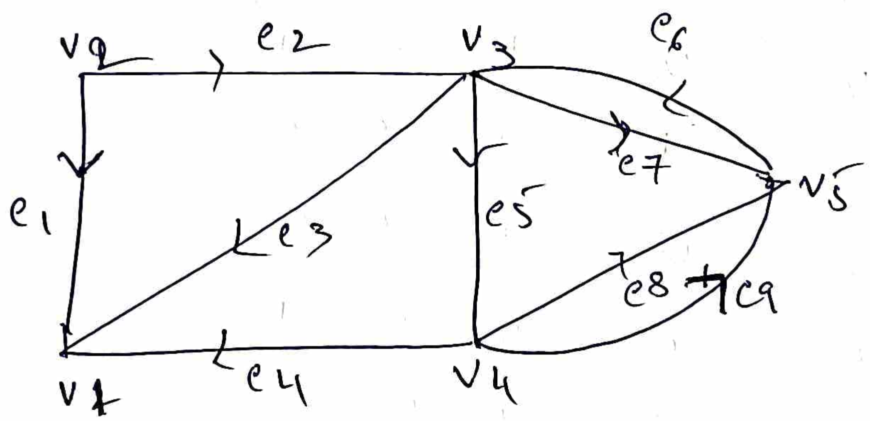
③ Incidence Matrix Representation (Directed graph)

Let $G=(V,E)$ is an directed graph with n vertices ordered from $v_1, v_2 \cdots v_n$ and m edges ordered from $e_1, e_2 \cdots e_m$ then the incidence matrix Representation of order $n \times m$ is denoted by I_G (or) $I(G)$.

The elements of the matrix $I_G = (M_{ij})_{n\times m}$ is defined as

$$M_{ij} = \begin{cases} +1, & \text{if } v_i \text{ is the initial vertex of edge } e_j \\ -1, & \text{If } v_i \text{ is the final vertex of edge } e_j \\ 0, & \text{otherwise} \end{cases}$$

eg: ①



$G(V,E)$

$$I_G = [M_{ij}]_{n\times m}$$

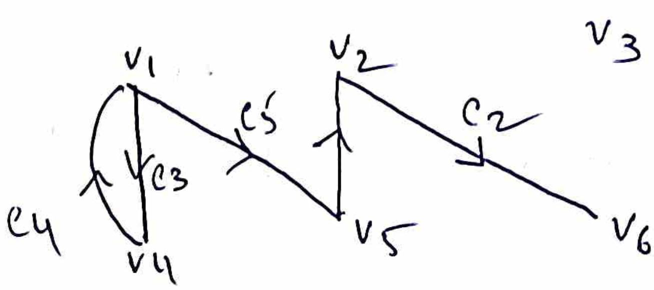
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & \end{array} \\ \left[\begin{array}{cccccccccc} -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & +1 & 0 & +1 & -1 & +1 & +1 & 0 & 0 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & +1 & +1 \\ 0 & 0 & 0 & 0 & 0 & +1 & -1 & -1 & -1 & - \end{array}\right] \end{array}$$

(2) Draw the graph whose incidence matrix is given below

$$\begin{bmatrix} 0 & 0 & 1 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

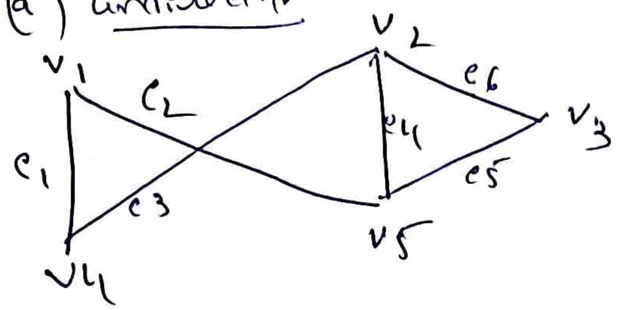
Ans :- Since the given matrix has 6 rows and 5 columns.
so its corresponding graph has 6 vertices and 5 edges.

$$\begin{array}{c|c} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \end{array}$$



(1) Find out the adjacency matrix and incidence matrix for the following graphs

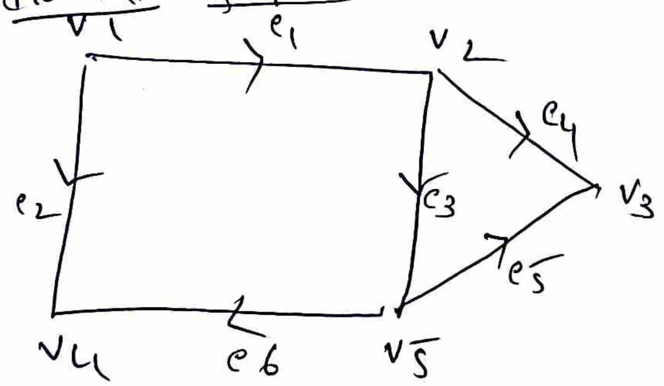
(a) undirected



Adjacency – vertex

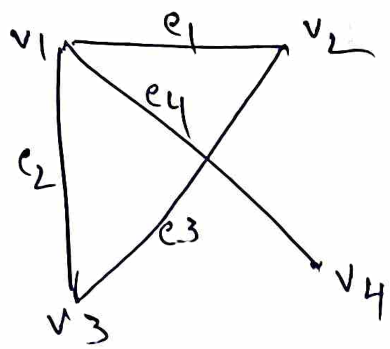
$$
\begin{array}{c c}
 & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} &
\begin{bmatrix}
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

(b) directed graph



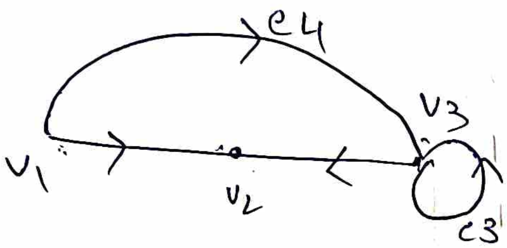
$$
\begin{array}{c c}
 & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} &
\begin{bmatrix}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

(3)



$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 1 & 1 \\ v_2 & 1 & 0 & 1 & 0 \\ v_3 & 1 & 1 & 0 & 0 \\ v_4 & 1 & 0 & 0 & 0 \end{array}$$

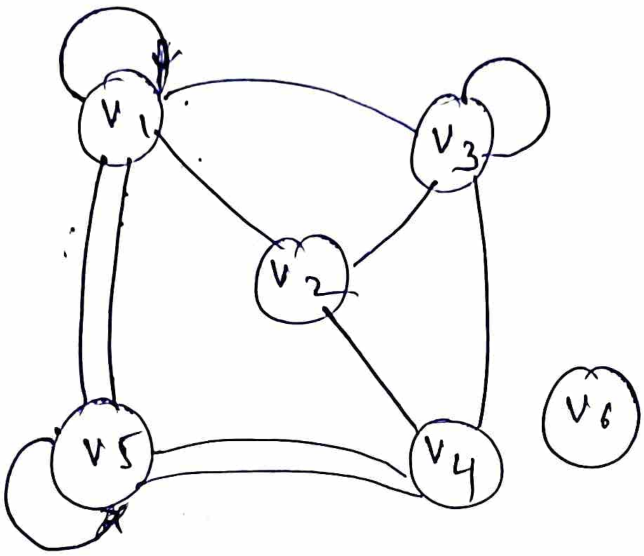
(4)

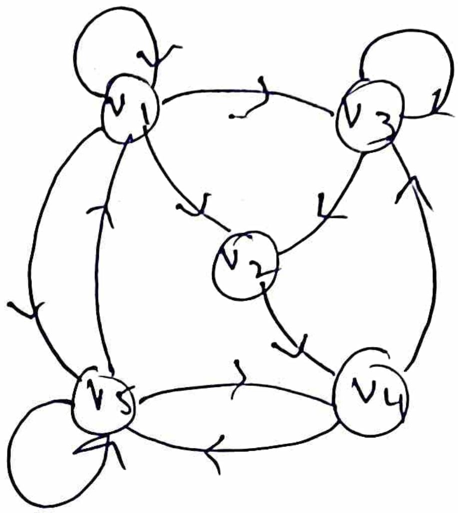


$$\begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & 0 & 1 & 1 \\ v_2 & 0 & 0 & 0 \\ v_3 & 0 & 1 & 1 \end{array}$$

Degrees of a vertex

undirected graph: | **Directed graph**





deg (v1) = 4+2 = 6
deg(v2) = 2
deg (v3) = 3+2 = 5
deg (v4) = 4
deg (v5) = 4+2 = 6
deg (v6) = 0

~~deg (v1) = 2~~

$\left.\begin{array}{l} v_1 = \text{Indegree}(v_1) = 1+1 = 2 \\ \quad \text{out}(v_1) = 3+1 = 4 \end{array}\right\} 6$

$\left.\begin{array}{l} v_2 = \text{Indegree}(v_2) = 2 \\ \quad \text{outdeg}(v_2) = 1 \end{array}\right\} 3$

$\left.\begin{array}{l} v_3 = \text{Indegree}(v_3) = 2+1 = 3 \\ \quad \text{outdeg}(v_3) = 1+1 = 2 \end{array}\right\} 5$

$\left.\begin{array}{l} v_4 = \text{In deg}(v_4) = 2 \\ \quad \text{out deg}(v_4) = 2 \end{array}\right\} 4$

$\left.\begin{array}{l} v_5 = \text{Indeg}(v_5) = 1+2 = 3 \\ \quad \text{out deg}(v_5) = 2+1 = 3 \end{array}\right\} 6$

ndegree :- No. of edges which one coming into the vertex.

out degree : No. of edges which one comming out.

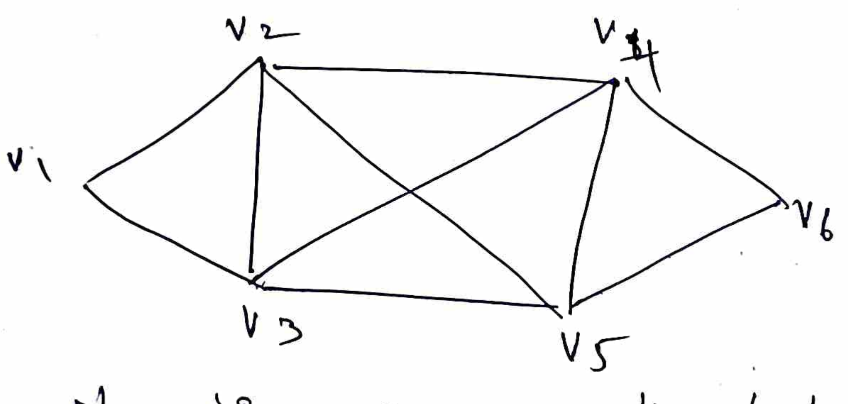
degree : The No. of edges associated with verteces.

loop : ① deg for undirected graph = 2

② deg for directed graph out + In deg

1 + 1

-) Find the degree of each vertex of the following graph.



Sol :- It is an undirected graph

dog (v_1) = 2

dog (v_2) = 4

dog (v_3) = 4

dog (v_4) = 4

dog (v_5) = 4

dog v_6 = 2

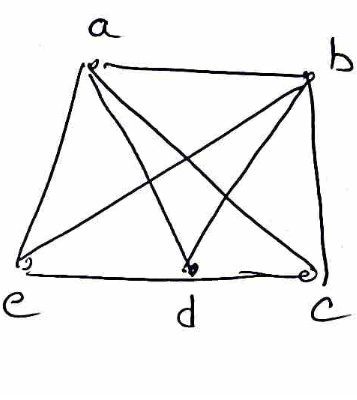
Planar graphs

Def:- A graph G is said to be planar graph if there exist some geometric representation of G which can be drawn on a plane such that no two of its edges intersect. The points of intersection are called cross-overs.

→ A ~~gra~~ drawing of a geometric representation of a graph on any surface such that no edges intersect is called embedding.

→ Note that if a graph has been drawn with crossing edges, it does not mean that it is nonplanar, there must be another way to draw the graph without cross-overs.

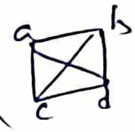
Eg:-



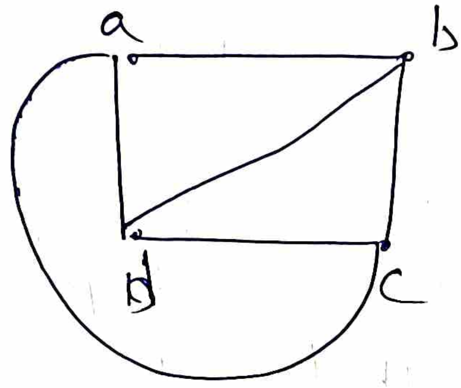
This graph is not planar because we have cross overs a, d, c, e.

-> Any graph without cross-overs is called a planar graph.

-> cross-overs :- Any edges should cross each-other then it is called cross-over.

{  }

Eg :- (2)



It is planar graph

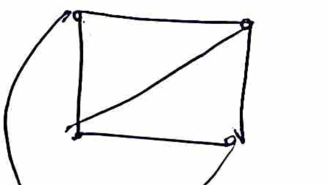
not have cross-overs.

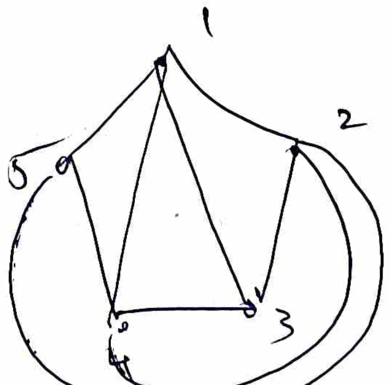
-> what ever we drawn on surface is called embedding

Eg : (1) K_3 - It is comple graph with 3 vertecy

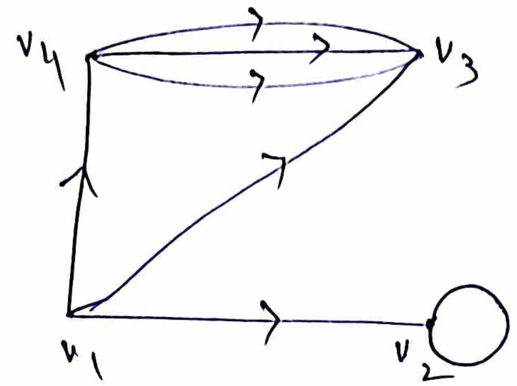
(3) K_5 - It is not planar graph

(2) K_4





-) Find the indegree, out degree and of total of each vertex of the following graph

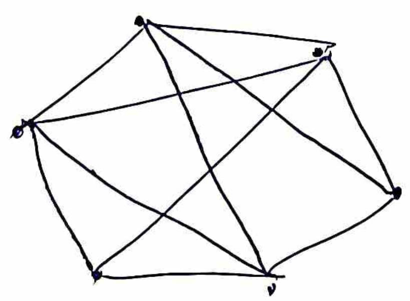


sol :- It is a directed graph

v_1 = Indeg = 0, out deg = 3, tot 0+3 = 3
v_2 = Indeg = 2, out deg = 1, tot 2+1 = 3
v_3 = Indeg = 4, out deg = 0, tot = 4
v_4 = Indeg = 1, out deg = 3, tot = 4

-) Does there exists a 4-regular graph on 6 vertices? If so construct a graph.

sol :- $q = \frac{p \times r}{2} = \frac{6 \times 4}{2} = \frac{24}{2} = 12$



Euler's formula (or) Euler's Theorem

A connected planar graph G with $|V|$ vertices and $|E|$ edges has exactly $|E| - |V| + 2$ regions in all of its diagrams.

$$|R| = |E| - |V| + 2$$

(or)

$$|V| - |E| + |R| = 2$$

} Euler's formula

where

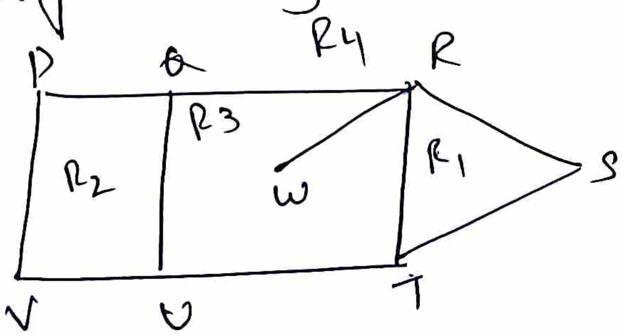
$|R|$ = no. of regions

$|V|$ = no. of vertices

$|E|$ = no. of edges.

The above formula is called Euler's formula.

Eg:- Show that the following graph is verified by Euler's formula



Sol :— The given graph G has

$|V| = 8$

$|E| = 10$

Euler's formula $|V| - |E| + |R| = 2$

$$|R| = |E| - |V| + 2$$
$$= 10 - 8 + 2$$
$$= 2 + 2$$
$$|R| = 4 \text{ Regions}$$

$R_1 = R - S - T - R$ (cycle)

$R_2 = P - Q - U - V - P$

$R_3 = Q - R - W - R - T - U - Q$

$R_4 = P - Q - R - S - T - U - V - P$

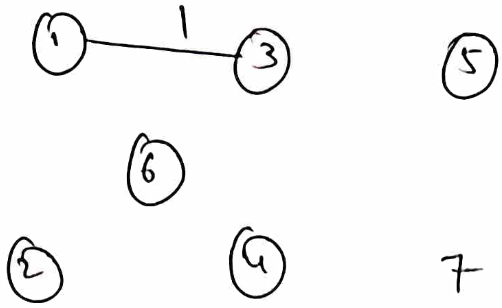
Euler's formula $|V| - |E| + |R| = 2$

$$8 - 10 + 4 = 2$$
$$-2 + 4 = 2$$
$$2 = 2 \quad \text{True}$$

$\therefore$ G satisfies the Euler's formula

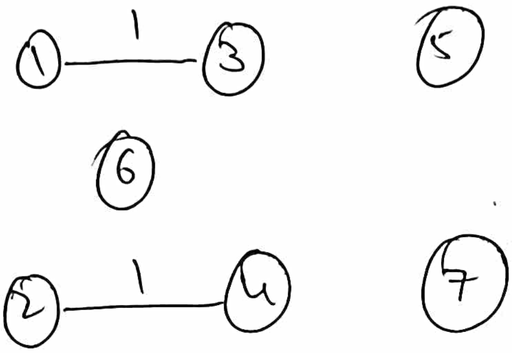
34 27

2



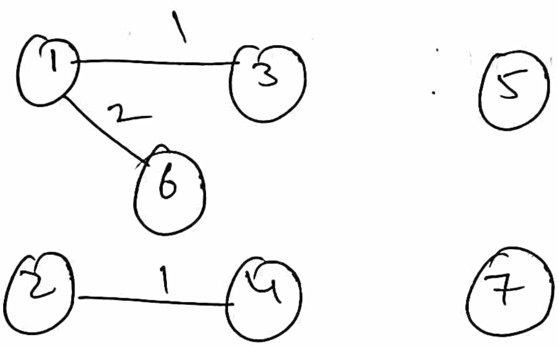
Menimum cost = 1

3



Menimum cost = 1+1 = 2

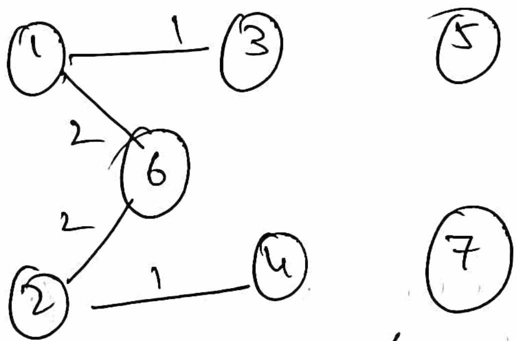
4



Menimum cost = 2

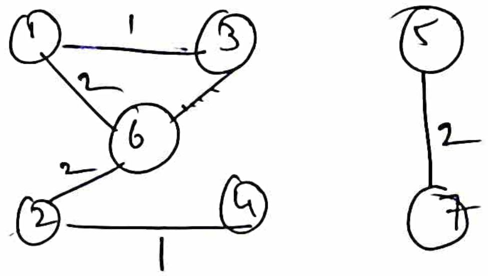
1+ 1+ 2 = 4

5



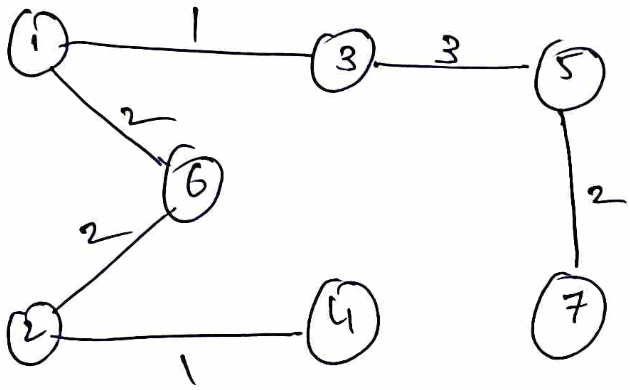
1+ 1+2+2 = 6
Menimum cost

(6)



Minimum cost = 1+1+2+2+2 = 8

(7)



Minimum cost = 1+1+2+2+2+3 = 11

(V = 7
E = 7-1 = 6
satisfies)

Algorithm kruskal's (E, cost, n, t)

E - set of edges
cost - cost of the c
n - no. of vertices
t - spanning tree

```
{
    i := 0;
    mincost = 0.
    while (i ≤ n-1)
    {
        Delete a minimum cost edge (u,v);
        J: = Find(u);
        k; = Find(V);

        If (J≠k)
        {
            i = i+1,
```

u —— v
1st vertex mincost u
2nd vertex mincost v

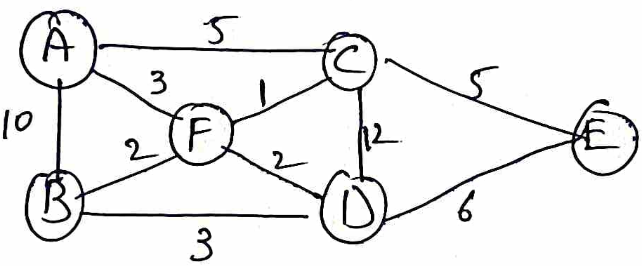
{ t[i:1] = u;
t[i:2] = v;
mincost = mincost + cost(u,v);
union (J,K);

}
}

cost(u,v)
u v
J K

this algorithm is used for finding out the min cost of spanning trees

(2) find min spanning trees by using Kruskal's Algorithm



Sol :- → Kruskal's algorithm is mainly used in order to calculate the min cost of spanning trees

→ Spanning tree means it is a graph of the given graph

→ Min cost of spanning tree means it produces min cost than the spanning trees

{ t[i : 1] = u ;
t[i : 2] = v ;

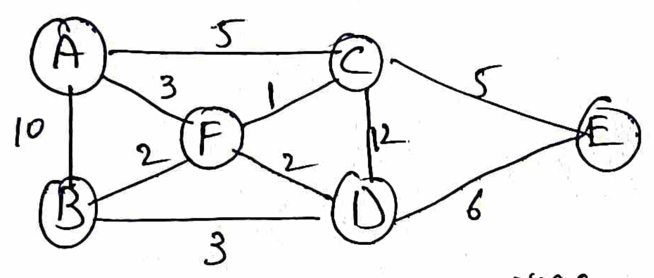
mincost = mincost + cost (u, v) ;
union (J, K) ;

cost(u,v)
u v
J K

}
}

this algorithm is used for finding out the min cost of spanning trees

(2)



sol: find min spanning trees by using Kruskal's Algorithm

sol :- → Kruskal's algorithm is mainly used in order to calculate the min cost of spanning trees

→ Spanning tree means it is a graph of the given graph

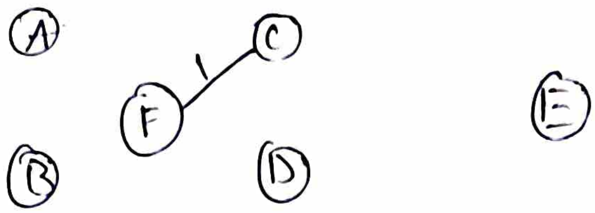
→ Min cost of spanning tree means it produces min cost than the spanning trees

Now we will construct min cost of spanning trees with the help of Kruskal's algorithm.

① First arrange all the edges in Ascending order means incremental order based upon the cost

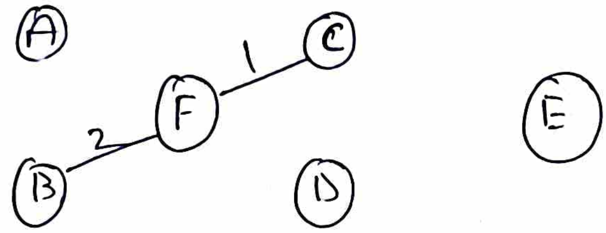
	edge	cost
①	C-F	1 ✓
②	B-F	2 ✓
③	D-F	2 ✓
~~④~~	~~C-D~~	~~2~~
⑤	A-F	3 ✓
⑥	B-D	3 ✗
⑦	A-C	5 ✗
⑧	C-E	5 ✓
⑨	D-E	6 ✗
⑩	A-B	10 ✗
⑪	C-D	12 ✗

1



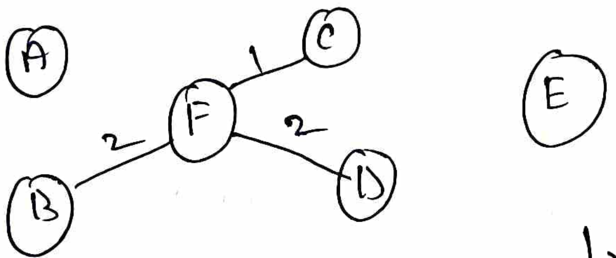
Min cost = 1

2



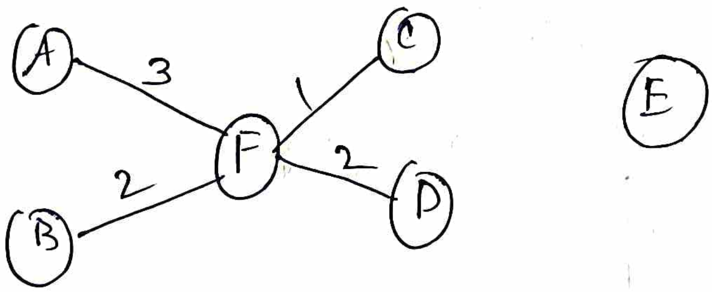
Min cost 1+2=3

3



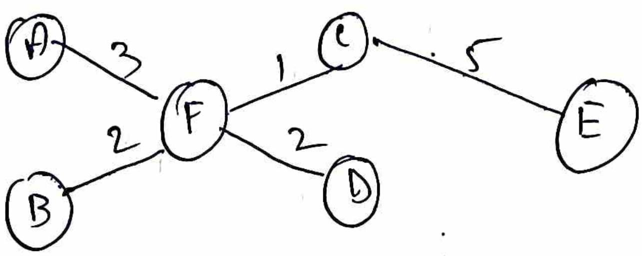
Min cost of spanning tree 1+2+2=5

4



min cost of S-T = 1+2+2+3=8

5



Min cost of S-T = 1+2+2+3+5

= 13

(1) The given graph contain all the vertices.

(2) graph contain n vertices then the spanning tree contains (n-1) edges

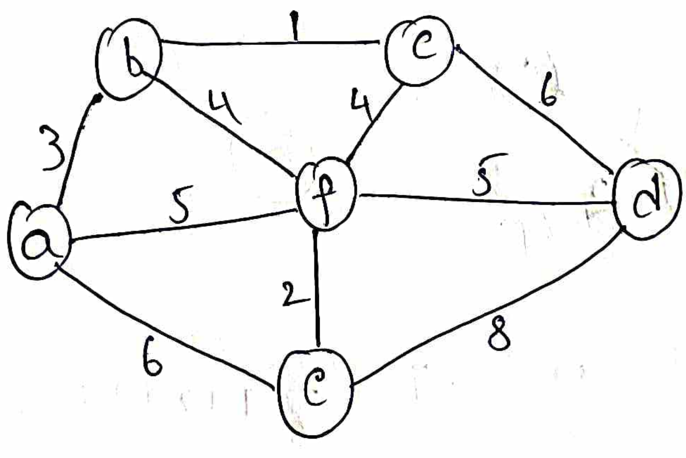
so G has n = 6

spanning tree contain n-1 = 6-1 = 5 edges

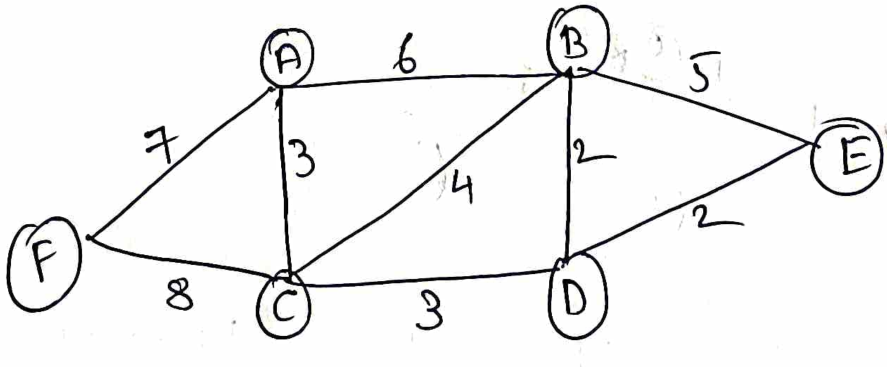
Also spanning tree should not contain any cycles.

H.W

(1) Find the minimum spanning trees by using kruskal's algorithm

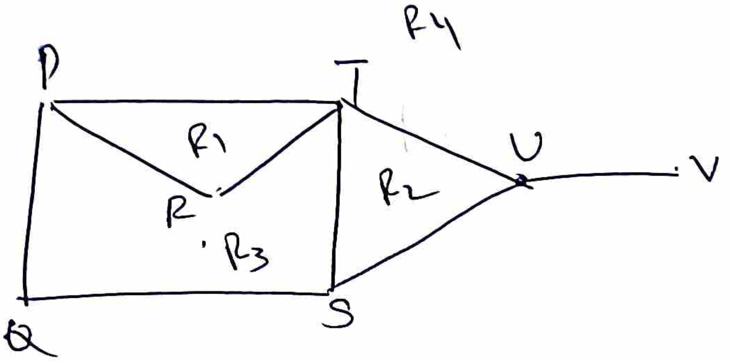


(2)



problem :

① For the given planar graph shown below, find the deg of each region and verify that sum of these degrees is equal to the twice the no. of edges. show That the following graph is verified by a Euler's formula.



Graph-G

Sol:- The given graph G has

$|V| = 7$, $|E| = 9$ edges

No. of regions can be calculated as

$$|V| - |E| + |R| = 2$$

$$|R| = |E| - |V| + 2$$
$$= 9 - 7 + 2$$
$$= 2 + 2$$
$$= 4 \text{ Regions}$$

compulsory we will check each region is a cycle.

degree of Region

$R_1 = P-T-R-P = 3$ (no. of edges be the degree in that R_1)

$R_2 = T-U-S-T = 3$ deg

$R_3 = P-R-T-S-Q-P = 5$

$R_4 = P-T-U-~~V~~-S-Q-P = 7$

Total degrees of all regions $3+3+3+7 = 18$

$$\sum_{i=1}^{4} \deg(R_i) = 2 * |E|$$

$$3+3+5+7 = 2*9$$

$$18 = 18 \text{ True}$$

Theorem is verified.

Euler's formula verification.

$$|V| - |E| + |R| = 2$$

$$7 - 9 + 4 = 2$$

$$-2 + 4 = 2$$

$$2 = 2 \text{ true}$$

∴ The given graph G satisfies the Euler's formula problem

(1) How many no. of regions are possible in a connected planar simple graph with 25 vertices each with a degree of 4 ?

sol :- no. of vertices in a connected planar simple graph $(V) = 25$

each vertex degree is $= 4$

$\therefore$ sum of degrees of all vertices in a given graph is $= 25 \times 4 \Rightarrow 100$

we know ~~that~~ sum of degrees of all vertices in a given graph $= 2 * |E|$

$$100 = 2 * |E|$$

$$|E| = \frac{100}{2} = 50$$

$\therefore$ No. of edges in a given graph is $|E| = 50$

from the Euler's formula we have to find out regions

$$|R| = |E| - |V| + 2$$

$$50 - 25 + 2$$

$$|R| = 27$$

$\therefore$ $|R| = 27$

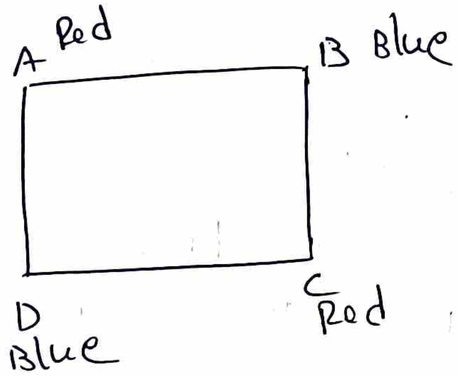
No. of regions in a graph $|R| = 27$

Imp — Graph coloring, Chromatic number

Given a planar or non-planar graph, If we assign colors to its vertices in such a way that no two adjacent vertices have the same color, then we can say That the graph G is properly colored.

proper coloring of a graph means assigning colors to its vertices such That adjacent vertices have different colors.

Eg:-

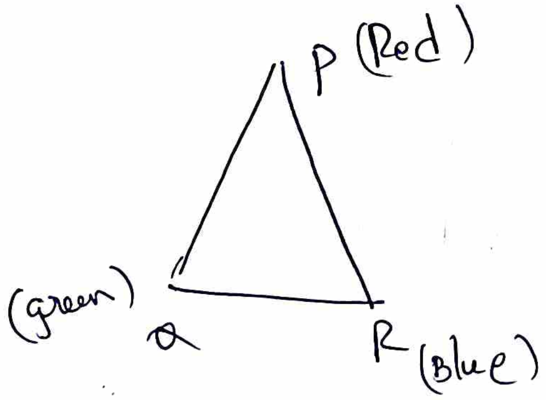


→ not have same color to the adjacent vertices

Graph-G.

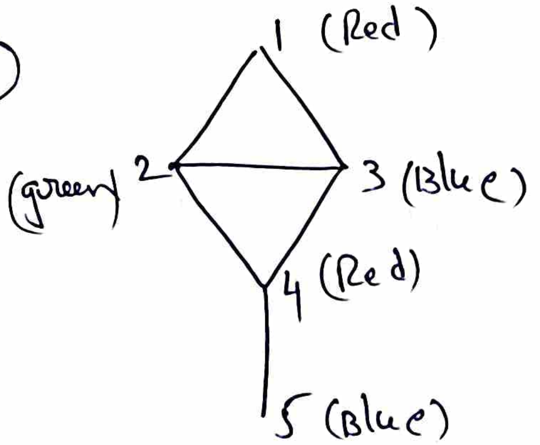
G is 2-colorable graph

(2)



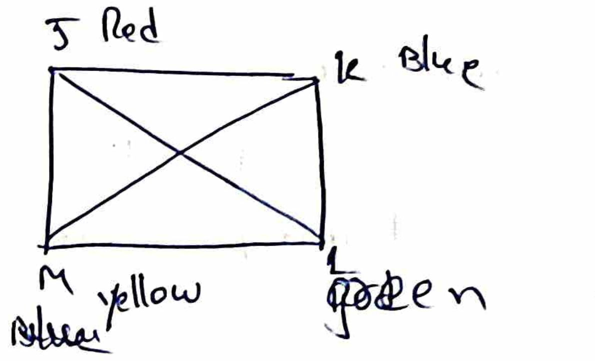
Graph is 3-colorable

3



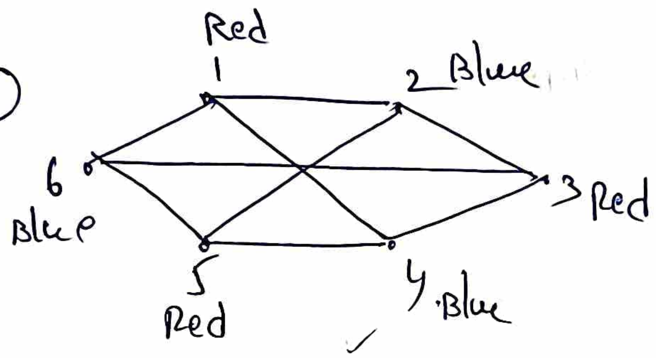
3 - colorable graph

4



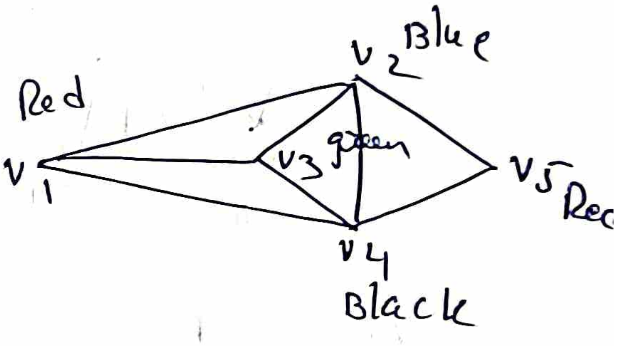
4 - colorable graph

5



2 - colorable graph

6



4 - colorable graph

chromatic number :

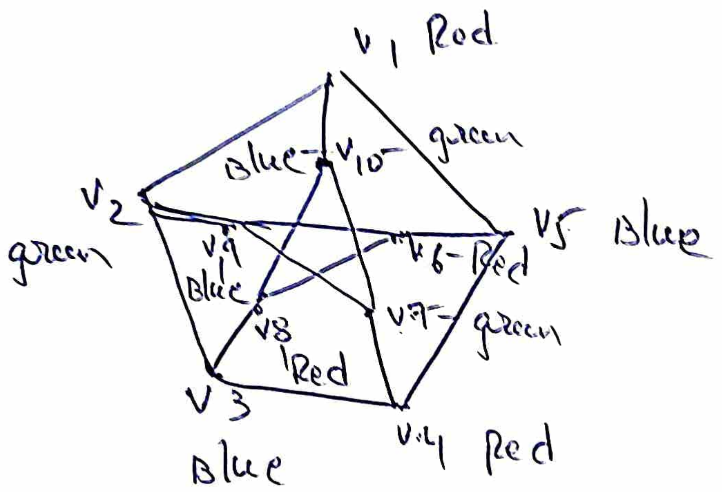
The minimum no. of colors required to color all the vertices of a given graph is called a chromatic number of a given graph.

The chromatic no. of a graph G is usually denoted by $\chi(G)$

A graph G is said to be K - colourable, If we can properly color it with k colors.

problems :-

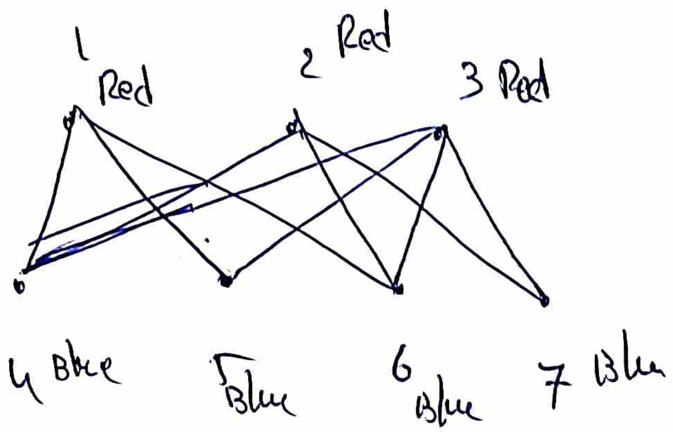
① Find the chromatic number of the following graphs.



V_1 - Red	V_6 - Red
V_2 - Green	V_7 - Green
V_3 - Blue	V_8 - Red
V_4 - Red	V_9 - Blue
V_5 - Blue	V_{10} - green

χ(peterson graph) = 3

Imp ②



$K_{3,4}$

~~Bog~~ Bi-partite graph means the graph is divided into two parts first ~~to~~ co-ordinate

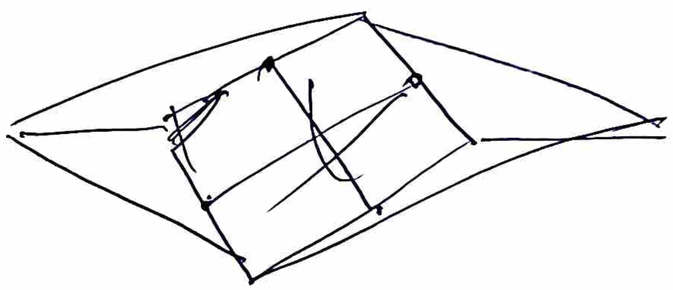
3 represents in the first graph.

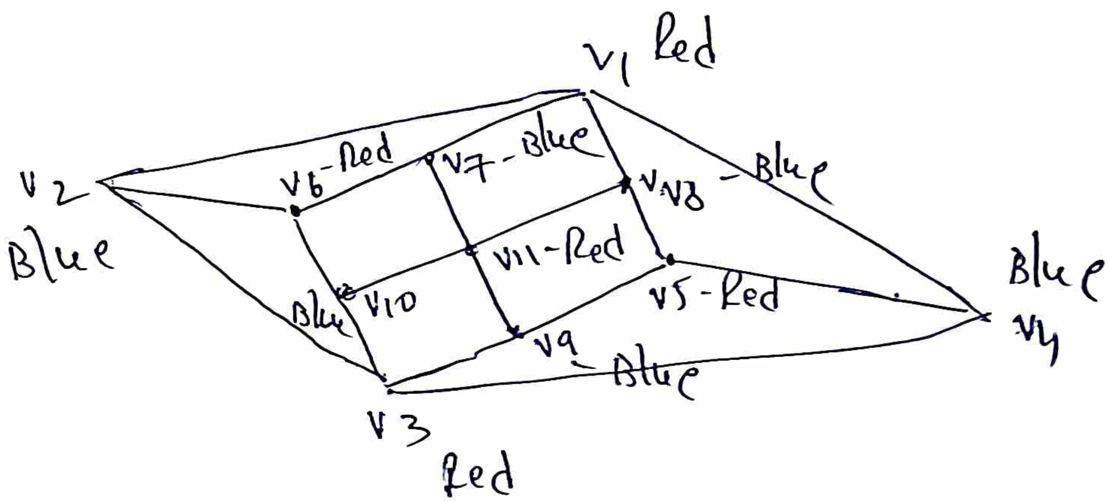
4 " " second graph

every vertex in the first part all connected in the second part then it is called complete- Bipartite graph.

$$\chi(K_{3,4}) = 2$$

(3)





v_1 - Red

v_2 — Bloe

v_3 — Red

v_4 — Blue

v_5 — Red

v_6 — Red

v_7 - Blee

v_8 — Blue

v_9 — Blue

v_{10} — Blue

v_{11} — Red

χ(Herschel graph) = 2

Imp

Kruskal's Algorithm

(33) (36)

the steps of the Algorithm are -

~~steps~~

~~① Arrange all the weights of the edges of the edges in increasing order of weights.~~

~~② First the minimum weight edge and added to the spanning tree~~

~~③ and add that edge~~

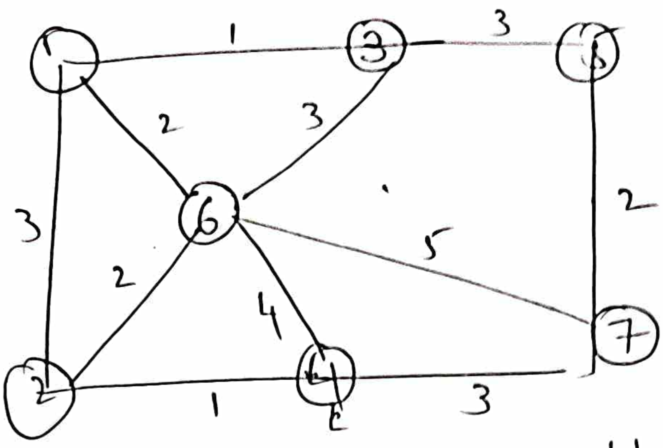
---

① Arrange all the edges in assending order based upon the cost

② select min cost edge from the list of sorted edges and add that edge to the tree if it is not forming any cycle, suppose if it forming any cycle then discard that edge

③ Repeate step 2 untill all the edges cover

(1)



Arrange the weights of the edges in G in increasing order

1 - 3 = 1 ✓

2 - 4 = 1 ✓ Accepted

1 - 6 = 2 ✓ Accepted

2 - 6 = 2 ✓ Accepted.

5 - 7 = 2 ✓

1 - 2 = 3 (It can be rejected due to forming cycle

3 - 6 = 3

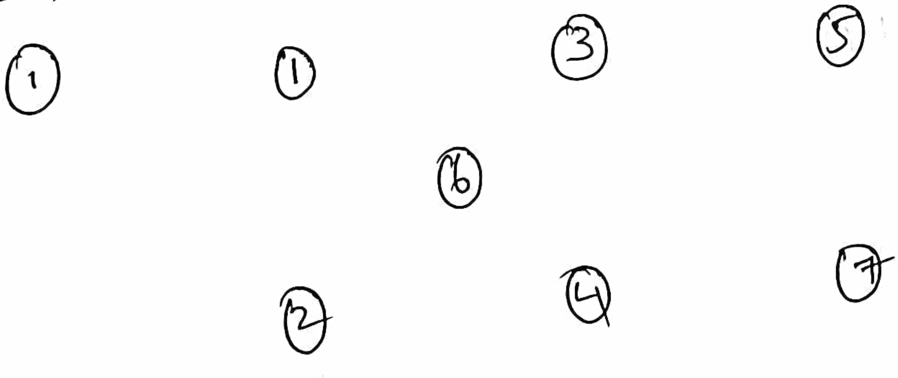
3 - 5 = 3 (Accepted)

4 - 7 = 3 ✗ (Rejected)

6 - 4 = 4 ✗ (")

6 - 7 = 5 ✗ (")

steps



Initially spanning tree is empty

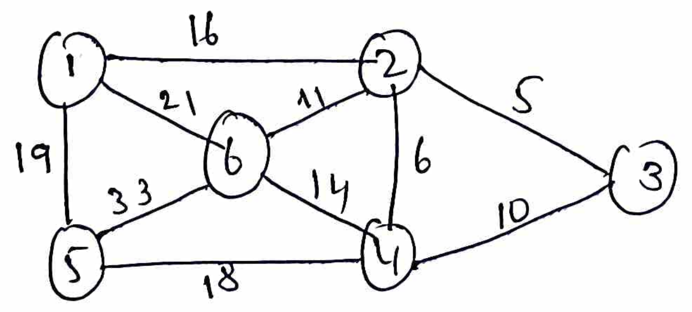
PRIM's Algorithm

steps (working Rule)

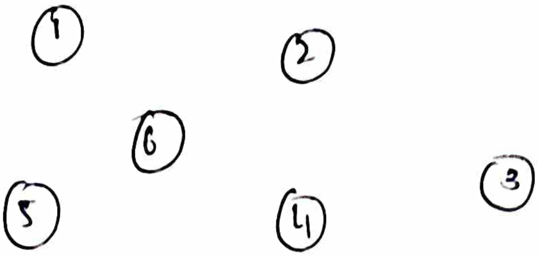
1. start with any vertex of the graph.
2. Find the edges associated with that vertex and add min cost edge to the spanning tree if it is not forming any cycle.
3. suppose it is forming any cycle discard that edge
4. we have to Repeat step 2 till all the vertices are covered.

problems

1. Find the minimum cost spanning trees by using prim's algorithm

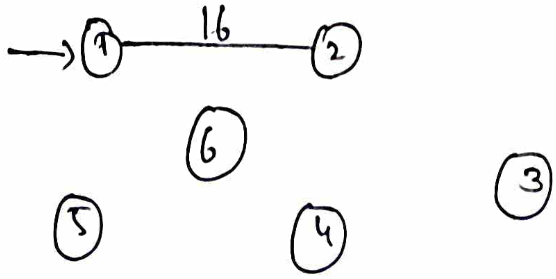


(1)



Min cost = 0

(2)



Min cost = 16

The neighbouring vertices of (1) are: (2), (6), (5).

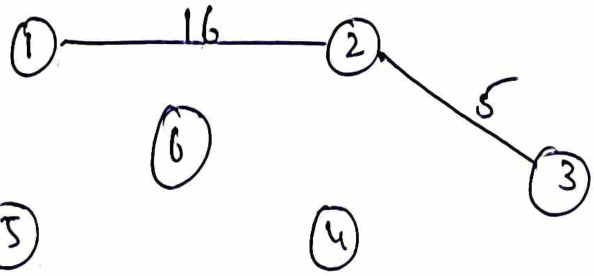
1-2 = 16 ✓

1-6 = 21

1-5 = 19

select 1-2 cost with cost 16

(3)



neighbours of vertex (2)

1-6 = 21

1-5 = 19

2-3 = 5 ✓

2-4 = 6

2-6 = 11

select ~~...~~ 2-3 edge so with cost 5, added to spanning tree without forming a cycle.

(4) adjacent neighbours of (1), (2), (3)

1-6 = 21

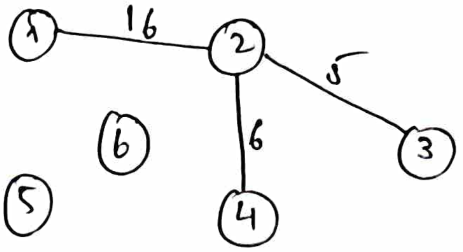
1-5 = 19

2-6 = 11

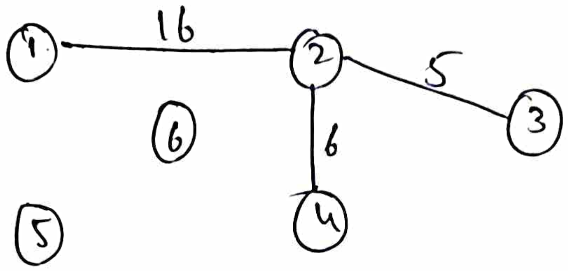
2-4 = 6 ✓

3-4 = 18 X

select 2-4 edge so with cost 6



5 Adjacent neighbours of 1, 2, 3 & 4



1-6 = 21

1-5 = 19

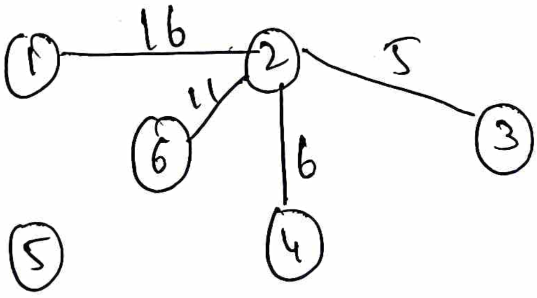
2-6 = 11 ✓ is min

4-6 = 14

~~3-4 = 10~~

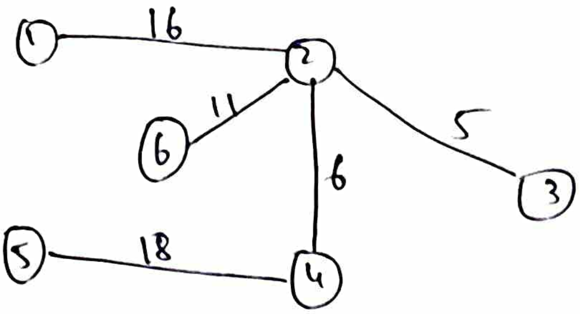
4-5 = 18

select edge 2-6, so with cost 11



Min cost of S-T = 16 + 5 + 11 + 6 = 38

(6)



Now check adjacent neighbours of 1, 6, 4, 3

1-5 = 19

1-6 = 21 ×

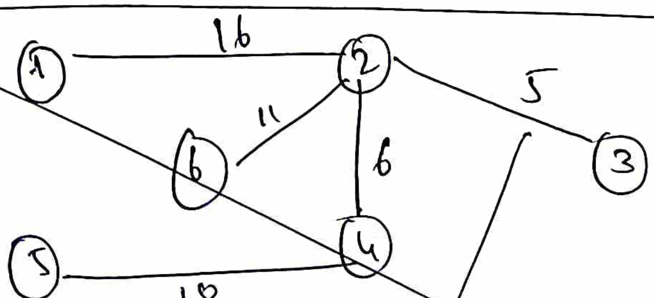
6-4 = 14 ×

4-3 = 10 ×

4-5 = 18 ✓ Min cost

Min cost = 16+11+6+18+5 = 56

(7)



Adjacent neighbours of 1, 6, 4, 3 & 5

1-5 = 19 ×

1-6 = 21 ×

5-6 = 33 ×

3-4 = 10 ×

4-6 = 14 ×

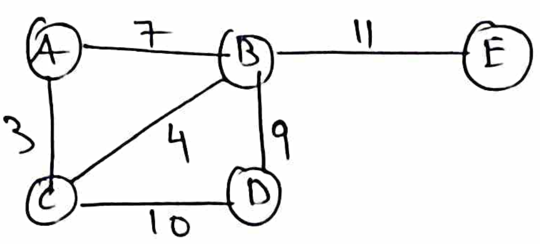
∴ It covered all the vertices

n = 6 vertices

S-Tree n-1 = 6-1 = 5 edges, not have cy

∴ =

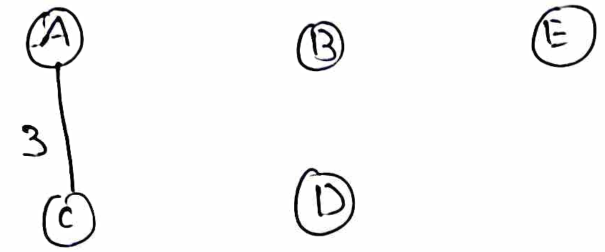
(2)



find min cost of spanning trees by prim's algorithm.

Sol :-

(1)



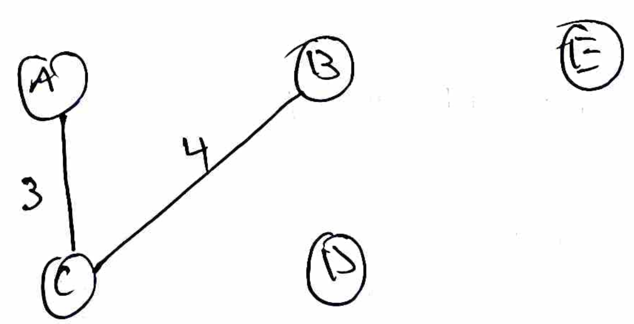
A-B = 7
A-C = 3 ✓ min cost

select A-C edge, with cost 3 so added to the S-T without forming a circle.

(2) find neighbours of (A), (C)

A-B = 7
C-B = 4 ✓
C-D = 10

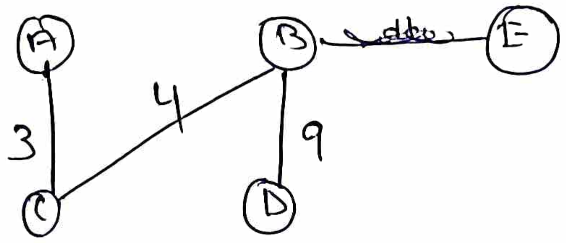
select C-B edge, with cost 4. added to the graph.



min cost = 3+4 = 7

(3) find neighbours of C, B

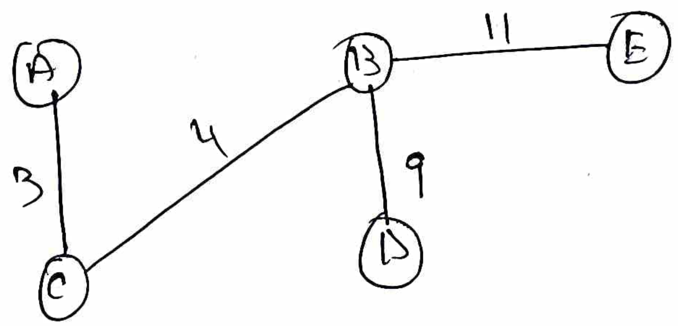
C-D = 10
D-B = 9 ✓ (min)
A-B = 7 ✗
B-E = 11



Min cost = 3+4+9 = 16

(4) Now find neighbours of A, B, D, C

A-B = 7 ✗
C-D = 10 ✗
B-E = 11 ✓ (min)

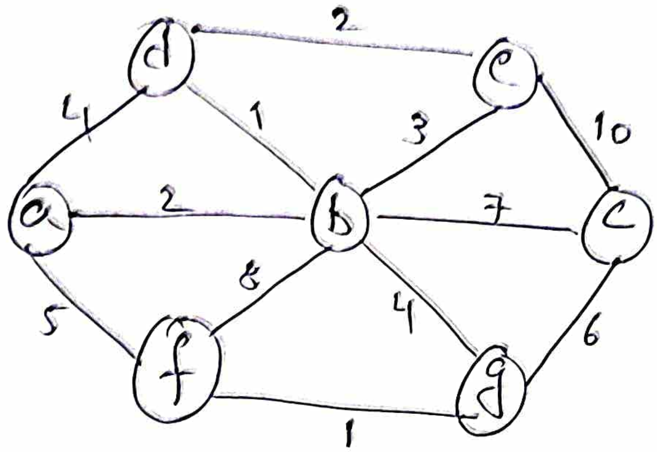


Min cost 3+4+9+11 = 27

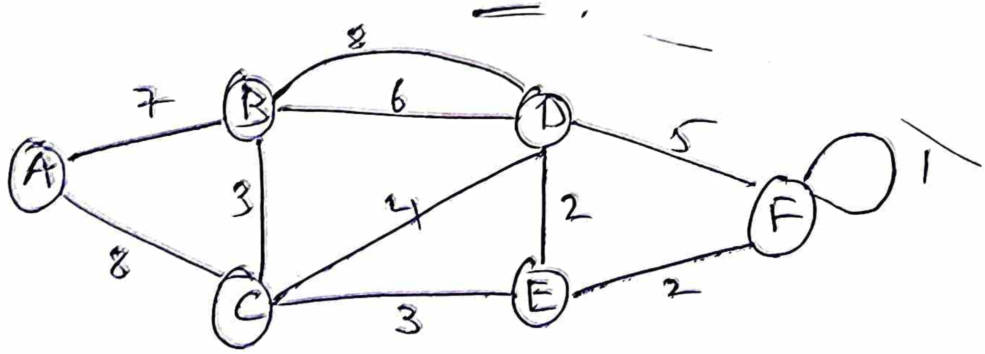
1. All edges covered.
2. graph contain n vertices, and have (n-1)e
n=5 vertices
S-T n-1 = 5-4 = 4 edges
Also does not contain any cycle.

H.W

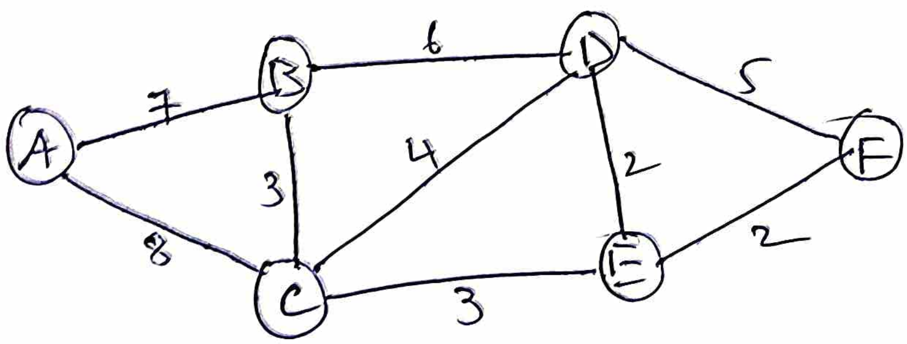
① find minimum spanning trees by using prims algorithm



②



Sol:- Remove loops & parallel edges, so the graph is

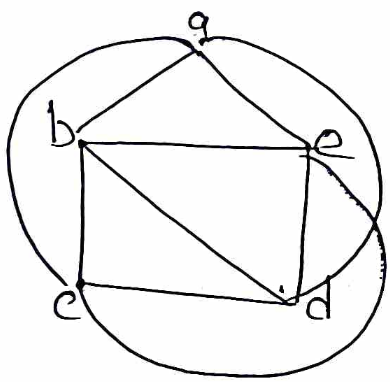


Imp Kuratowski's graph

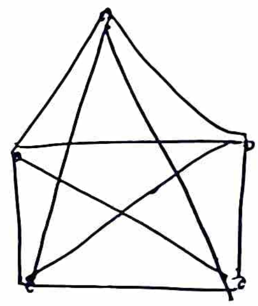
two specific non-planar graphs are called Kuratowski's graph.

→ The complete graph with 5 vertices (K_5) is 1st Kuratowski's graph.

sug:- (complete graph means all the vertices connected to each other. It is denoted by K_n)



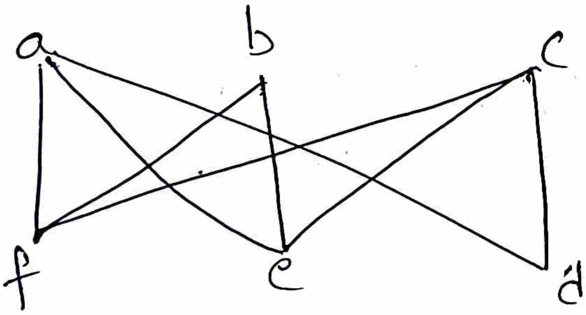
(or)



K_5

The second graph is a regular connected graph with 6 vertices and 9 edges ($K_{3,3}$) ~~[illegible]~~ ($K_{m,n}$)

It is comple Bi-partite graph.



($K_{3,3}$)

This is Kuratowski's graph.

Both are non-planar graphs.

Note :

① K_5 & K_{33} are connected non planar graphs

② K_5 and K_{33} are regular graphs

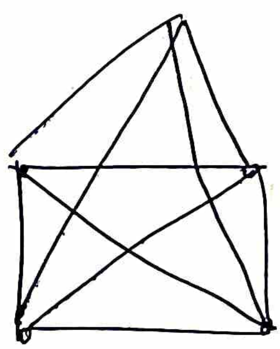
③ K_5 is a non-planar graph with smallest no. of vertices.

④ $K_{3,3}$ is a non-planar graph with smallest no. of edges.

⑤ Deletion of a vertex or an edge makes K_5 and $K_{3,3}$ a planar graph.

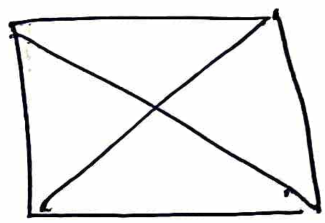
Kuratowski's Theorem :- A graph is planar iff it does not contain K_5 or $K_{3,3}$ as a subgraph. In other words, A graph is planar iff it has no sub-graph homeomorphic to K_5 or $K_{3,3}$

① Show that $K_5 - v$ (delete one vertex) is planar.



K_5

$K - v \to$



In K_5, $n=5$, $e = \frac{n(n-1)}{2} = \frac{5(5-1)}{2} = 10$

In $K_5 - v$, $n=4$, $e = \frac{4(4-1)}{2} = 6$

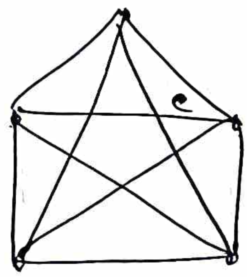
Since $K_5 - v$ is, K_4 contains circuit of length 3

$\therefore$ $e \le 3n-6 \Rightarrow 6 \le 3.4-6 = 6$ which is true

Hence Kuratowski's first graph is planar if one vertex is removed from that graph.

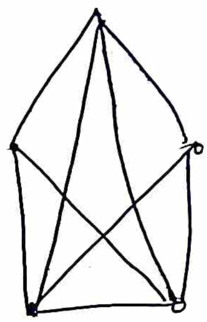
(2) Show that $K_5 - e$ is planar.

Sol :-



K_5

$K_5 - e$



In K_5, $n=5$, $e=10$

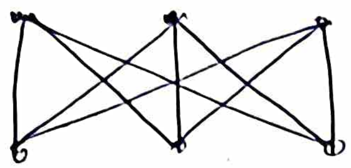
$K_5 - e$, $n=5$, $e = 10-1 = 9$

$\therefore$ from Theorem $e \le 3n-6$

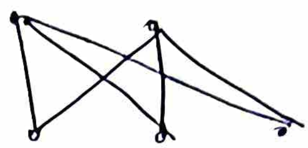
$\Rightarrow 9 \le 3.5-6 = 9$ which is true

Hence Kuratowski's first graph is planar after removing one edge from the graph.

(3) S.T $K_{3,3}-v$ is planar.



$(K_{3,3})$



$(K_{3,3}-v)$

In $K_{3,3}$, $n=6$, $e = 3\times3 = 9$

In $K_{3,3}-v$, $n=5$, $e = 2\times3 = 6$

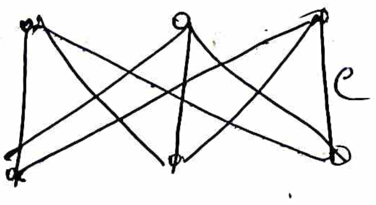
Since $K_{3,3}-v$ is $K_{2,3}$ does not contain any circuit of length 3,

$\therefore$ from the Theorem $e \le 2n-4$

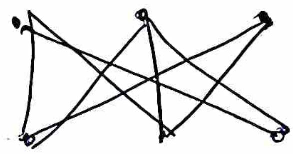
we get $6 \le 2\times5-4 = 6$ which is true.

Hence, Kuratowski's second graph is planar if one vertex is removed from the graph.

(4) S.T $K_{33}-e$ is planar



$(K_{3,3})$



$(K_{33}-e)$

In $K_{3,3}$, $n=6$, $e=9$

$K_{3,3}-e$, $n=6$, $e=9-1=8$

since $K_{3,3}-e$ does not contain any circuit of length 3,

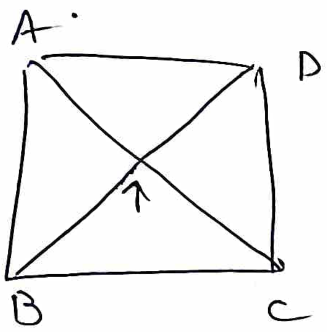
∴ from the theorem $e \le 2n-4$

we get $8 \le 2\times 6-4=8$ which is true

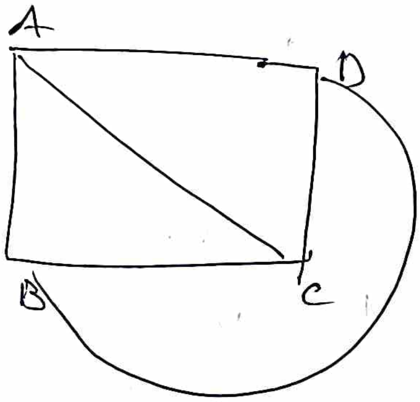
Hence Kuratowski's second graph is planar if remove one edge from the graph $K_{3,3}$

Eg: <u>Planar graph</u>

(1)

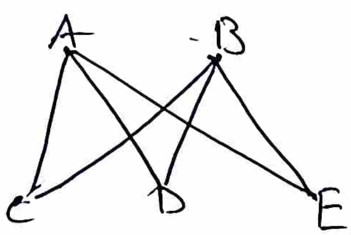


complete graph K_4

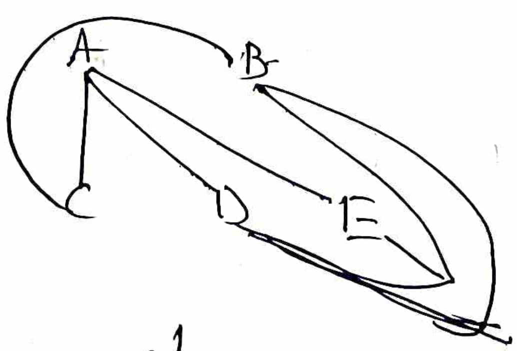


planar graph

(2)

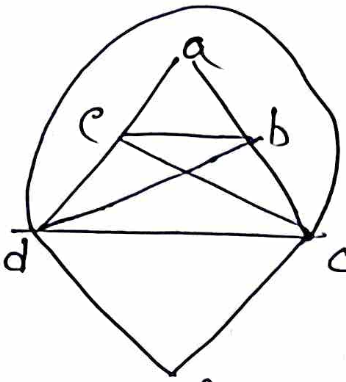


Bipartite graph

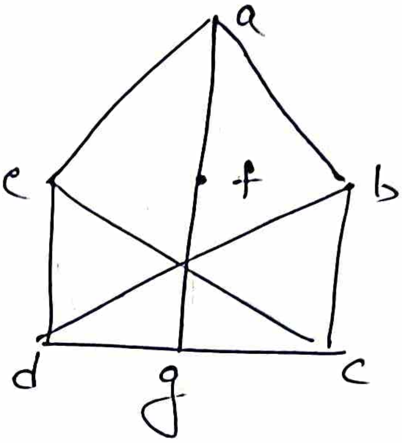


planar graph.

② Show that the following graphs are non-coplanar



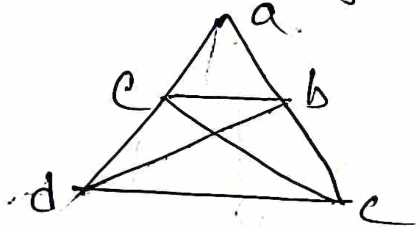
(i)



(ii)

Sol :-

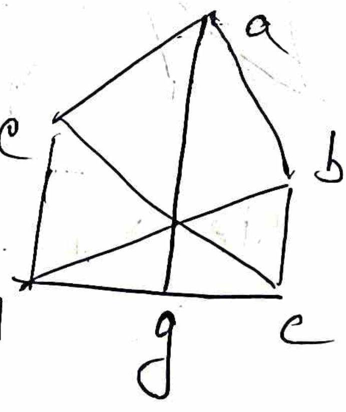
(i) Deleting the vertex f and the edges incident with it from the given graph we get the subgraph



This subgraph is isomorphic to K_5

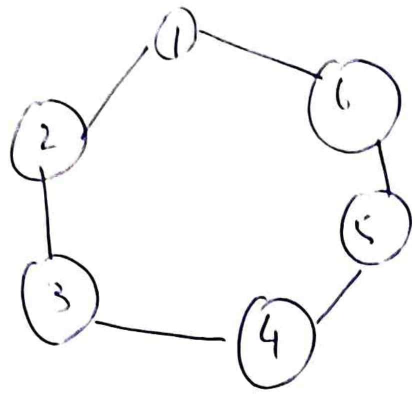
using Kuratowski's theorem the given graph is non-planar.

(ii) deleting the vertex f on the path a-f-g we get the subgraph



This subgraph is nothing but $K_{3,3}$. using Kuratowski's theorem the given graph is non-planar

Minimum cost of spanning trees



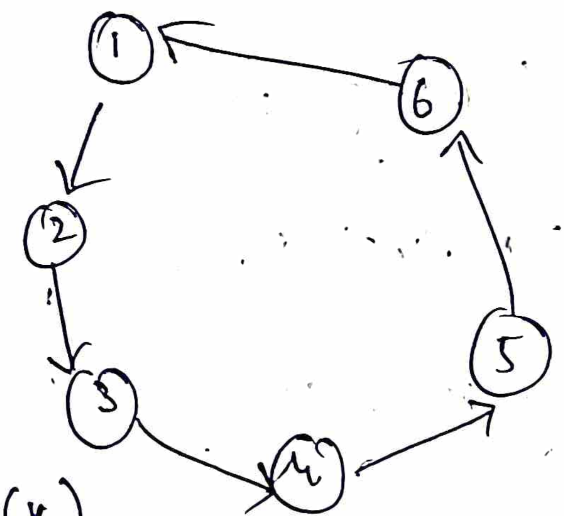
G (V, E)

$V = \{1, 2, 3, 4, 5, 6\}$

$E = \{(1,2), (2,3), (3,4), (4,5), (5,6), (6,1)\}$

Each and every edge is undirected graph

->



— It is directed graph.

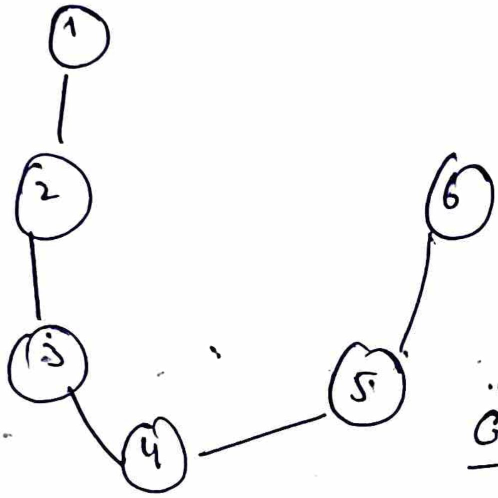
G'(V)

A subgraph G'(V, E') is a spanning t[ree] iff G' is a tree. It satisfies the following conditions

1. $G'(V) = G(V)$

2. $G'(E) = \cancel{G(E)}\ G(V) - 1$

$\therefore$ G' is a subgraph, T

-> 

G'

$V = \{1, 2, 3, 4, 5, 6\}$

$E = |V| - 1$

$E = 6 - 1 = 5$ (main graph i.e. G)

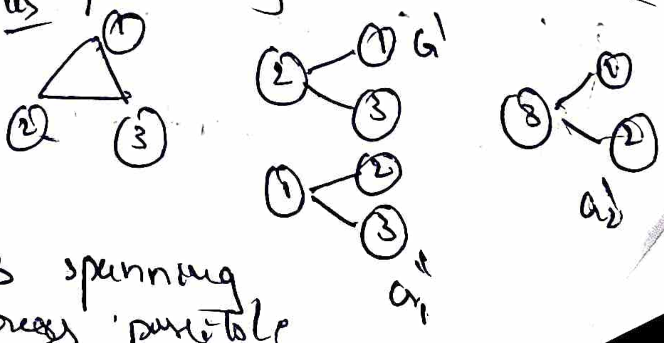
$= \{(1,2), (2,3), (3,4), (4,5), (5,6)\}$

G' is subgraph, so it is a spanning tree.

-> How many no. of ~~us~~ spanning trees are possible?

say :- n^{n-2}

eg :-



$n = 3$

$3^{3-2} = 3$ spanning trees possible