GRAPH THEORY

1. Representation of graph
2. DFS
3. BFS
4. Spanning trees
5. planar graph

Graph :-

Graph is a collection of vertices and edges. it can represent in the form of $G = (V, E)$

Where V = vertices
E = Edges

Representation of graph :-

A graph 'G' and its contain set of vertices and Edges. It can be classified into 3 types.

1. Matrix representation
2. Adjacency matrix
   Incidence Matrix
3. Linked list representation

Matrix representation :-

The matrix are commonly used to represent graphs for computer processing.

a) Adjacency matrix :-

Here we have to represent

i) Representation of undirected graph
ii) Representation of directed graph.

i) Representation of undirected graph :-

The adjaceny matrix of a graph 'G' with 'n' vertices and no parallel edges is an 'n' by 'n' matrix

$$A = \{a_{ij}\}$$

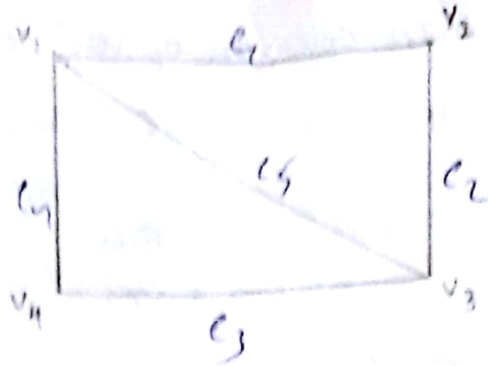
Where

$$a_{ij} = \begin{cases} 1, & \text{if there is an edge b/w } i^{th} \text{ \& } j^{th} \text{ vertices} \\ 0, & \text{if there is no edge b/w } i^{th} \text{ and } j^{th} \text{ vertices.} \end{cases}$$

$$A(i,j) = \begin{cases} 1 \text{ for } (i,j) \in E \\ 0 \text{ for } (i,j) \notin E \end{cases}$$

Scanned by CamScanner

Eg:-



In above graph it contains 4 vertices i.e., v_1, v_2, v_3, v_4 and edges e_1, e_2, e_3, e_4, e_5. The required adjacent matrix

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

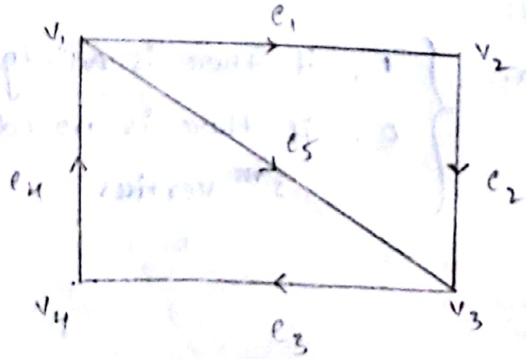
ii) Representation of directed graph :-

The adjacency matrix of a directed graph 'G' with 'n' vertices matrix

$$A = \{a_{ij}\}_{n \times n}$$

$$A = \begin{cases} 1, & \text{if arc } (v_i, v_j) \text{ is in } G \\ 0, & \text{if arc } (v_i, v_j) \text{ is not in } G \end{cases}$$

Eg:-



In the above directed graph contain 4 vertices and 5 edges the required adjacency matrix of directed graph is

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Incedence Matrix :-

Here we have to represent

a) Representation of undirected graph

b) Representation of directed graph

Representation of undirected graph :-

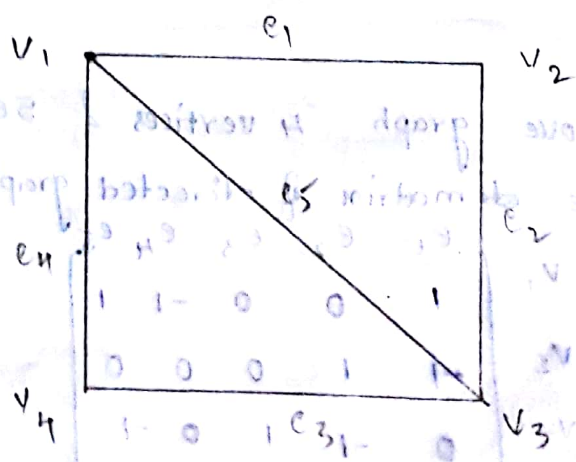
Consider a undirected graph $G = (V, E)$ which has 'n' vertices and 'm' edges the incedence matrix

$$B = \{b_{ij}\}_{n \times m}$$

Where

$$b_{ij} = \begin{cases} 1, \text{ When edge } e_j \text{ is incedent with vertices } v_i \\ 0, \text{ When edge } e_j \text{ is not incedent with vertices } v_i \end{cases}$$

Eg:-



In above undirected graph contain 4 vertices and 5 edges therefore the required incedence matrix of undirected graph is

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

adje

Representation of directed graph :-

The incedence matrix $B = \{b_{ij}\}_{n \times m}$

where n = vertices

m = edges

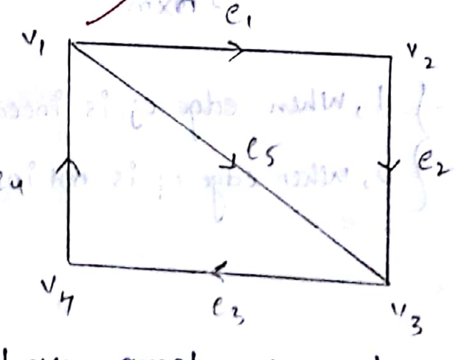
ii)

Where

$$b_{ij} = \begin{cases} 1, & \text{if arc } j \text{ is directed away from a vertex } v \\ -1, & \text{if arc } j \text{ is directed toward vertex} \\ 0, & \text{Otherwise} \end{cases}$$

'G'

Eg:-



In the above graph 4 vertices & 5 edges the required incedence matrix of directed graph is

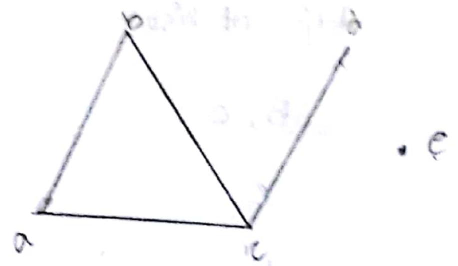
$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \end{array}$$

Linked - List Representation :-

In this representation a list of vertices adjacent to each vertices is maintained. This representation is also called "adjacency structure representation." In this we have to represent directed graphs and undirected graph.

Un-directed graph :-

Eg:-

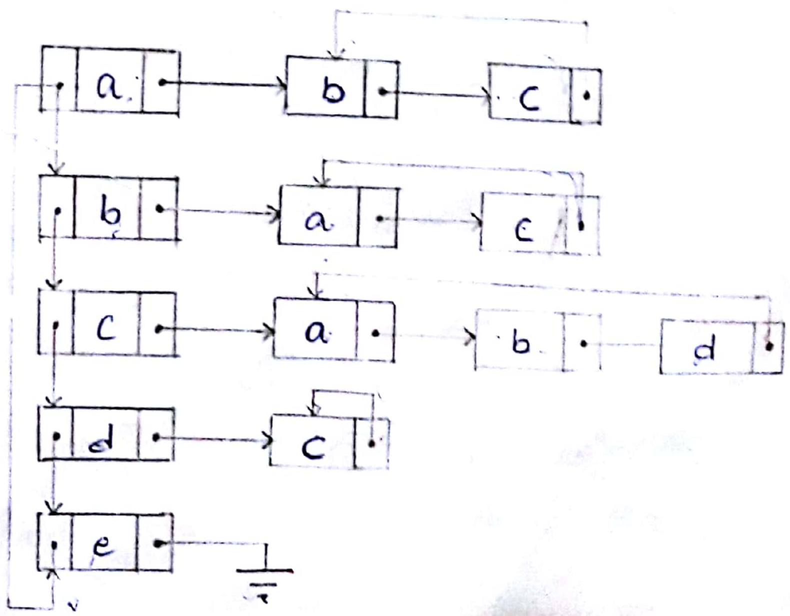


In above graph contain 5 vertices and 4 edges

The required adjecent list is

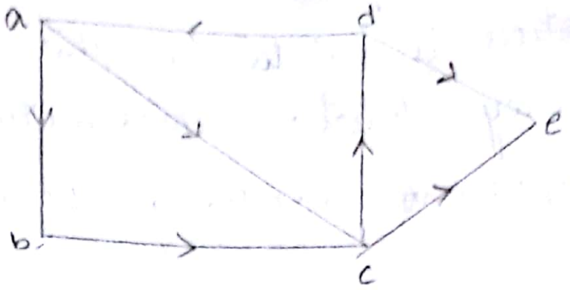
Vertex	adjecent list
a	b, c
b	a, c
c	a, b, d
d	c
e	ϕ

The linked list representation of undirected graph is



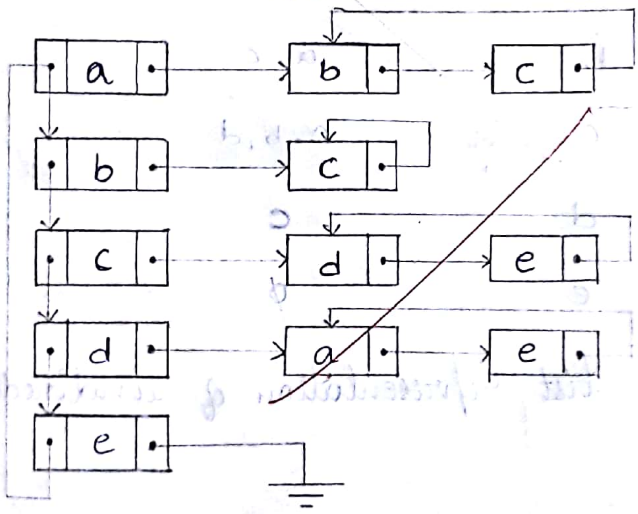
Directed graph :-

Eg:-

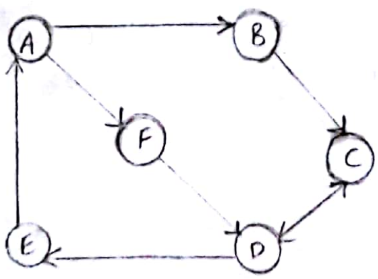


Vertex	Adjecent list
a	b, c
b	c
c	d, e
d	a, e
e	ϕ

The linked list representation of directed graph.



Eg:-



Tree :-

A tree is a connected acyclic graph i.e., a connected graph having no cycles. It's edges are called branches.

Eg:- 1)

Vertices = n = 1

Edges = n-1 = 0

2) 

Vertices = n = 2

Edges = n-1 = 1

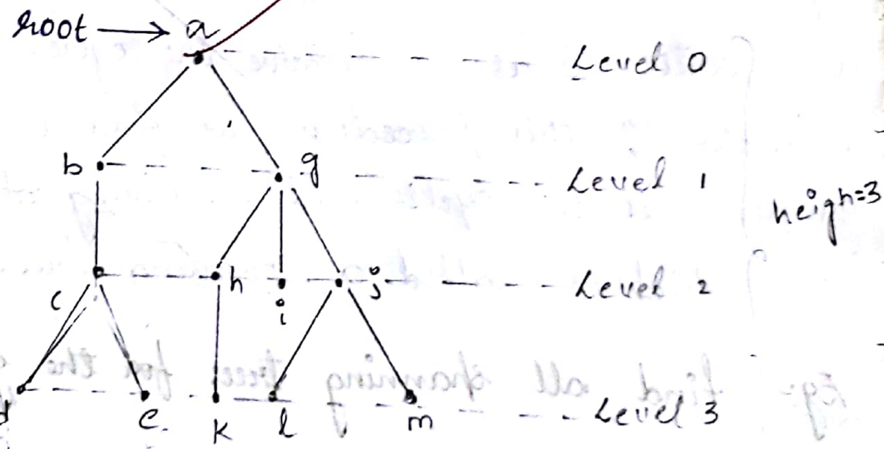
3)  4)  5)  6) 

1) 'G' is connected graph and has no cycle.

2) 'G' is connected and has 'n-1' edges.

3) 'G' is acyclic and has 'n-1' edges

4) There is exactly one path between every pair of vertices in 'G'.



* D is child of C
* C is parent of D
* D & e are siblings

Sibling :- Two vertices that are both children of the same parent are called sibling.

→ If the vertex has no children, then that vertex is called leaf node (or) terminal vertex.

Eg:- In above tree D, e, k, l, m are leaf nodes (or) terminal vertices.

Spaning Tree :-

A sub graph 'H' of a graph 'G' is called a Spanning tree of 'G', if

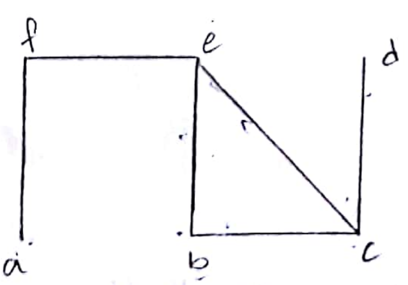
i) 'H' is tree and

ii) 'H' contain all vertices of 'G'

A spanning tree i.e, directed tree is called a directed spanning tree of 'G'.

* if 'G' is a connected graph and that contains cyclis, Then we can delete any edge, with out delting any vertex so as to remove the cycle. By repeated application of this procedure, we find a connected sub graph with out cycles that containing all the vertices of 'G' which is called a spanning tree.

Eg:- find all spanning trees for the graph 'G'



Sol A Graph 'G' is connected, it has 6 Edges and 6 vertices.

Hence each spanning tree must have 'n-1' edges

i.e., $= 6-1$

$= 5$ edges

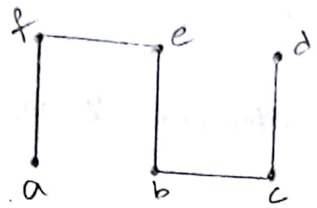
And $m-n+1$ edges are deleted from 'G'

$= m-n+1$ $\qquad$ m = edges

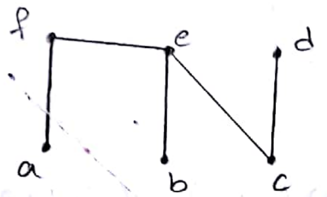
$= 6-6+1$ $\qquad$ n = vertices

$= 1$.

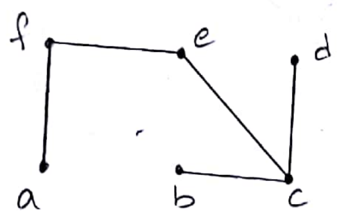
i) Delete (e, c) edge



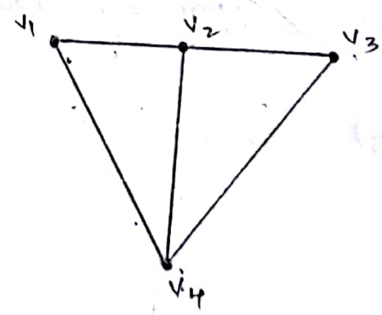
ii) Delete (b, c) edge



iii) Delete (b, e) edge.



Eg:-



Note:- Let 'G' be a connected simple graph with $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ and $m = (m_{ij})_{n\times n}$ is a matrix in which

m_{ii} = Deg(v_i) and $m_{ij} = -1$, when v_i and v_j are adjcent. $m_{ij} = 0$ other wise.

Then the no. of spanning trees of 'G' is co-factor of element of m

$$
\begin{array}{cc}
 & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
m = \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}
\end{array}
$$

The co-factor of the element 2 (m_{11}) is the determinant

$$
D = \begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix}
$$

$m - n + 1$
$5 - 4 + 1$
$1 + 1$
2

$= 3(6-1) - (-1)(-3-1) + (-1)(1+2)$

$= 3(5) + (-4) - (3)$

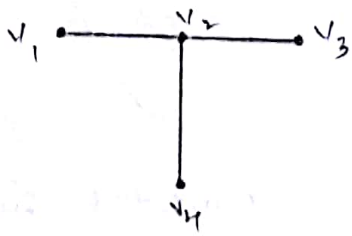
$= 15 - 4 - 3$

$= 15 - 7$

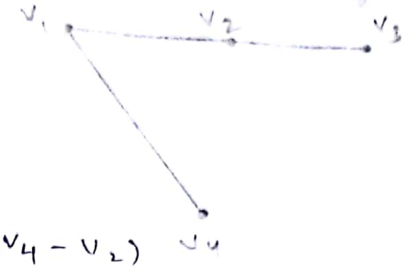
$= 8.$

Thus there are 8 spanning trees.

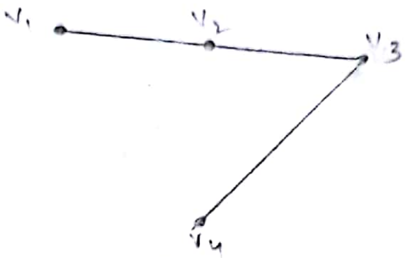
i) $(v_1 - v_4)$, $(v_4 - v_3)$



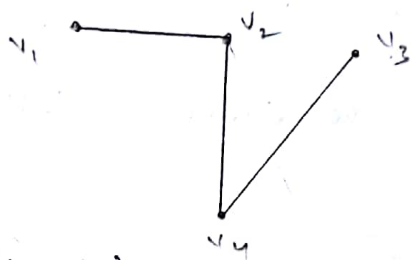
ii) $(v_2 - v_4)$ $(v_4 - v_3)$



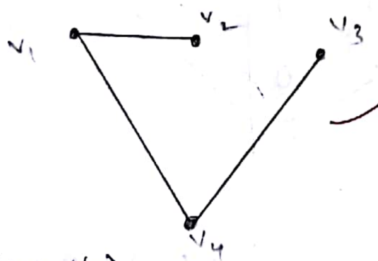
iii) $(v_1 - v_4)$ $(v_4 - v_2)$



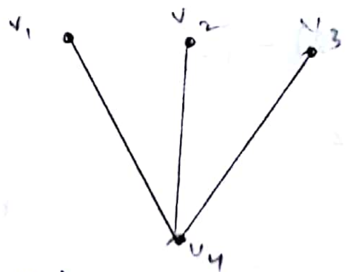
iv) $(v_1 - v_4)$ $(v_2 - v_3)$



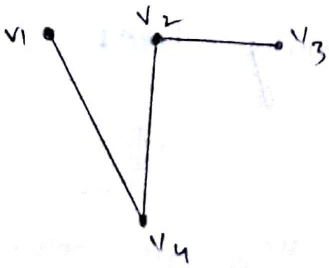
v) $(v_2 - v_3)$ $(v_2 - v_4)$



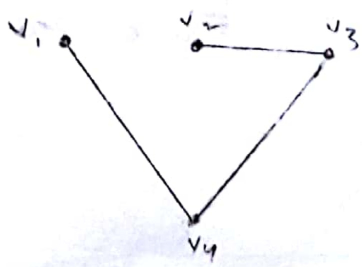
vi) $(v_1 - v_2)$ $(v_2 - v_3)$



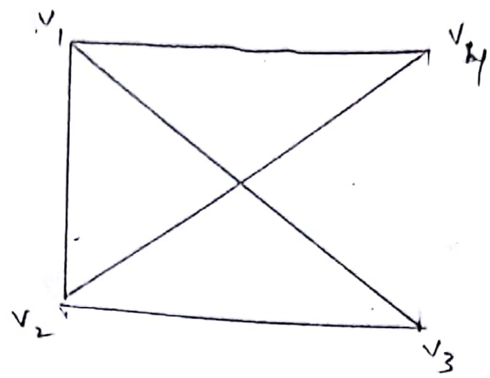
vii) $(v_1 - v_2)$ $(v_4 - v_3)$



viii) $(v_1 - v_2)$ $(v_2 - v_4)$



Q. How many spanning trees has the graph G



$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 3 & -1 & -1 & -1 \\ v_2 & -1 & 3 & -1 & -1 \\ v_3 & -1 & -1 & 2 & 0 \\ v_4 & -1 & -1 & 0 & 2 \end{array}$$

5-4+1

(2)

The co-factor of the element $3\,(m_{11})$ is the determinent

m-n+1

5-4+1

1+1

2

$$D = \begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{vmatrix}$$

$= 3(4-0) - (-1)(-2-0) + (-1)(0+2)$

$= 3(4) + (-2) - (2)$

$= 12 - 2 - 2$

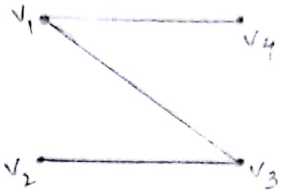
$= 12 - 4$

$= 8$

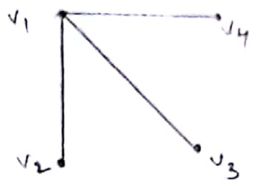
$\therefore$ There are 8 spanning trees

i) Delete $(v_1 - v_3)$ $(v_1 - v_4)$ edges

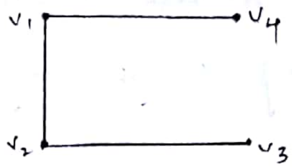
ii) $(v_2 - v_1)$ $(v_2 - v_4)$ edges



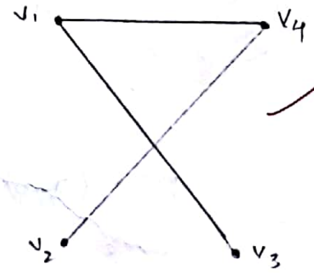
iii) $(v_2 - v_3)$ $(v_2 - v_4)$ edges



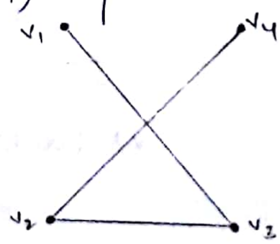
iv) $(v_1 - v_3)$ $(v_2 - v_4)$ edges



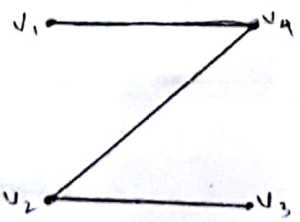
v) $(v_1 - v_2)$ $(v_2 - v_3)$ edges



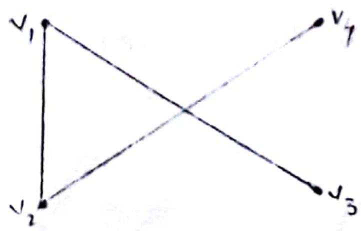
vi) $(v_1 - v_2)(v_1 - v_4)$ edges



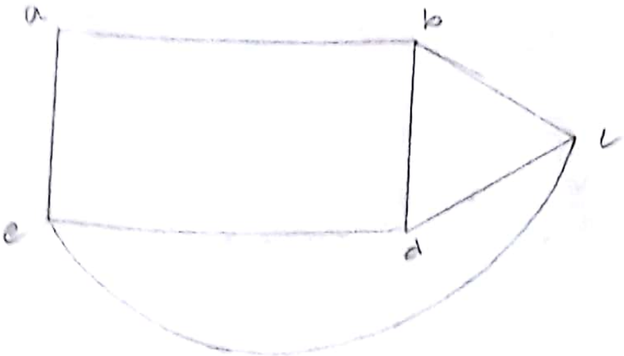
vii) $(v_2 - v_1)$ $(v_1 - v_3)$



viii) $(v_1 - v_4)$ $(v_2 - v_3)$



4. Find 6 spanning trees of the graph G



Sol

In the above graph contains 7 edges, 5 vertices

$V = 5$, $E = 7$

$\therefore$ $m - n + 1$

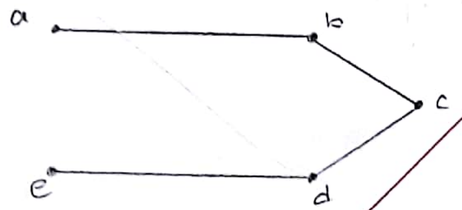
$= 7 - 5 + 1$

$= 2 + 1$

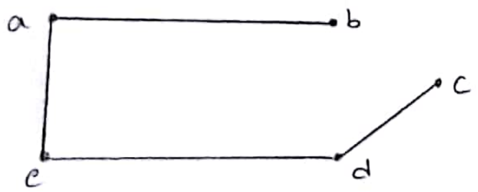
$= 3$ Edges are deleted

We give only following 6 spanning trees.

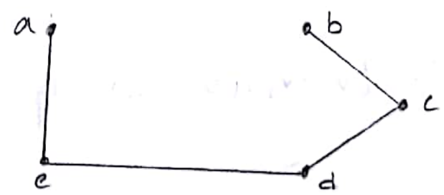
i) (a-e) (e-c) (b-d)



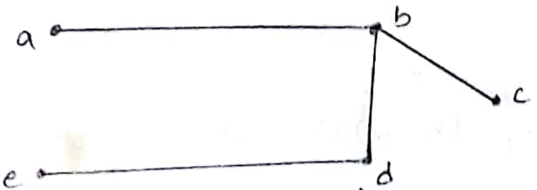
ii) (b-c) (b-d) (c-e)



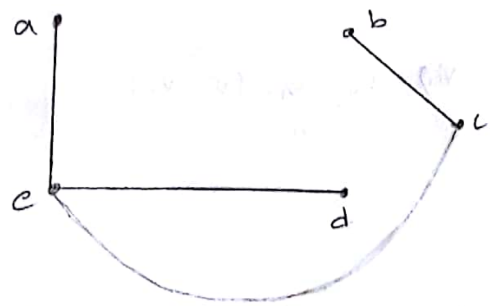
iii) (a-b) (b-d) (c-e)



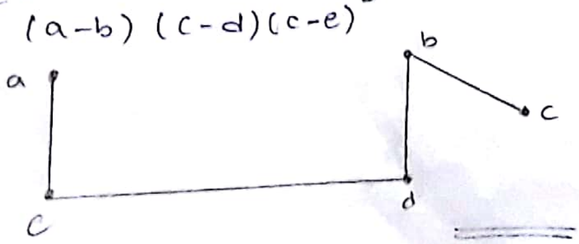
iv) (a-e) (c-d) (c-e)



v) (a-b) (b-d) (d-c)

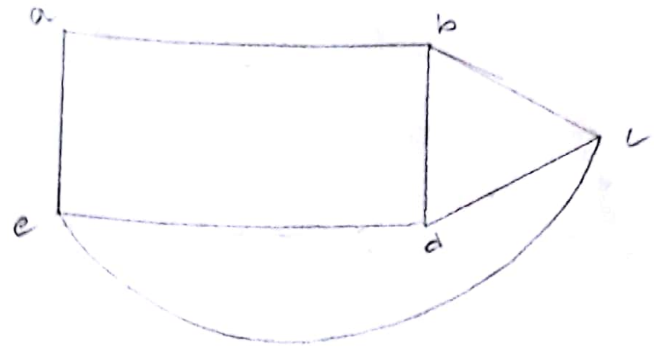


vi) (a-b) (c-d) (c-e)



Scanned by CamScanner

4. Find 6 spanning trees of the graph G



Sol

In the above graph contains 7 edges, 5 vertices

$V = 5$, $E = 7$

$\therefore \; m - n + 1$

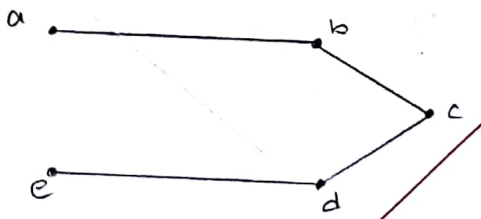
$= 7 - 5 + 1$

$= 2 + 1$

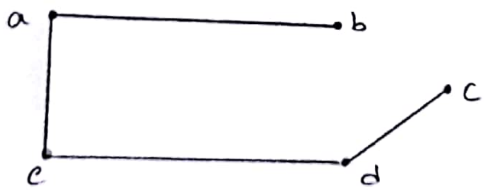
$= 3$ Edges are deleted

We give only following 6 spanning trees.

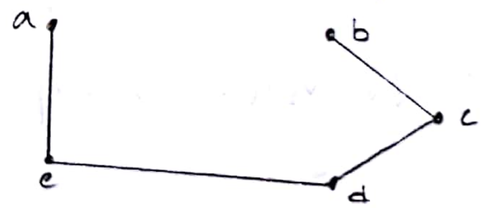
i) (a-e) (e-c) (b-d)



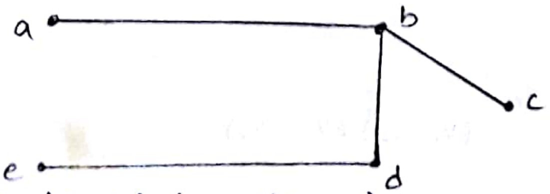
ii) (b-c) (b-d) (c-e)



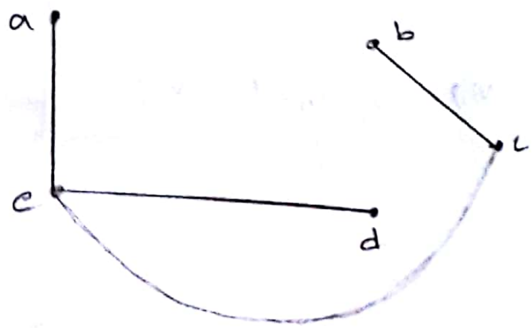
iii) (a-b) (b-d) (c-e)



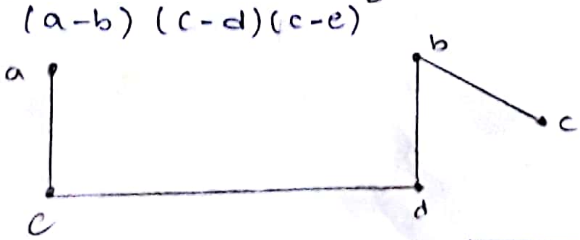
iv) (a-e) (c-d) (c-e)



v) (a-b) (b-d) (d-c)

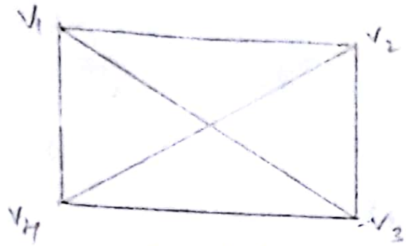


vi) (a-b) (c-d) (c-e)



Find all spanning trees of K_4?

Sol



graph K_4.

$$
\begin{array}{c|cccc}
 & V_1 & V_2 & V_3 & V_4 \\
\hline
V_1 & 3 & -1 & -1 & -1 \\
V_2 & -1 & 3 & -1 & -1 \\
V_3 & -1 & -1 & 3 & -1 \\
V_4 & -1 & -1 & -1 & 3
\end{array}
$$

$m - n + 1$
$= 6 - 4 + 1$
$= 2 + 1$
$= 3$

The co-factor of the element 3 (m_{11}) the determinent

$$D = \begin{vmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{vmatrix}$$

$= 3(9-1) - (-1)(-3+1) + (-1)(1+3)$

$= 3(8) + (-3-1) - (1+3)$

$= 24 - 4 - 4$

$= 24 - 8$

$= 16$

A complete graph k_n has n^{n-2} passible spanning tree

$K_4 = 4^{4-2}$

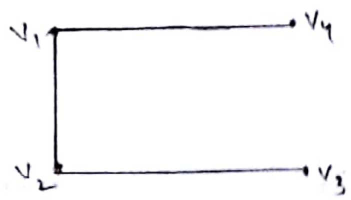
$= 4^2$

$= 16$

4.

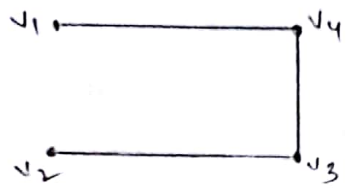
i) $(v_1 - v_3)\ (v_2 - v_4)\ (v_1 - v_4)$

ii) $(v_3 - v_4)\ (v_1 - v_3)\ (v_2 - v_4)$

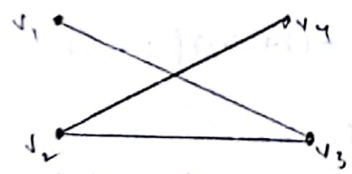


iii) $(v_2 - v_3)\ (v_2 - v_4)\ (v_3 - v_1)$

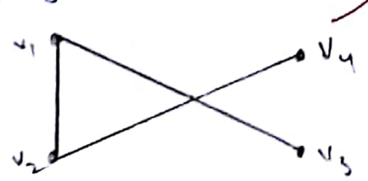
iv) $(v_1 - v_2)\ (v_1 - v_3)\ (v_2 - v_4)$



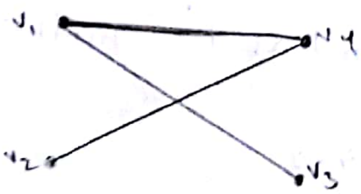
v) $(v_1 - v_2)\ (v_4 - v_3)\ (v_1 - v_4)$



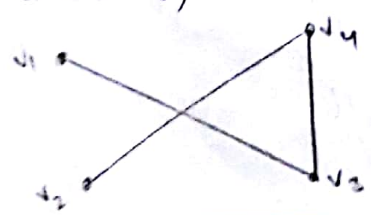
vi) $(v_1 - v_4)\ (v_4 - v_3)\ (v_3 - v_2)$



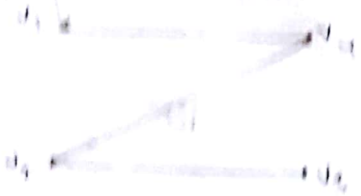
vii) $(v_1 - v_2)(v_2 - v_3)\ (v_3 - v_4)$



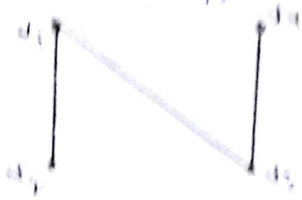
viii) $(v_4 - v_1)\ (v_1 - v_2)\ (v_2 - v_3)$



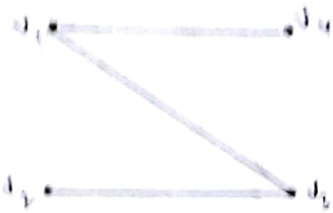
$(v_1 - v_2)(v_2 - v_3)(v_3, v_4)$



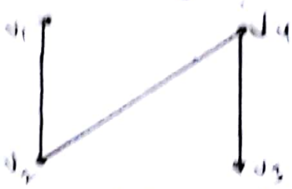
v) $(v_1 - v_4)(v_1 - v_3)(v_1 - v_4)$



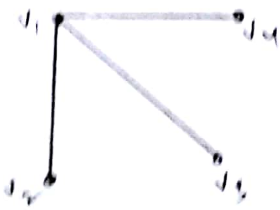
) $(v_1 - v_3)(v_2 - v_4)(v_3 - v_4)$



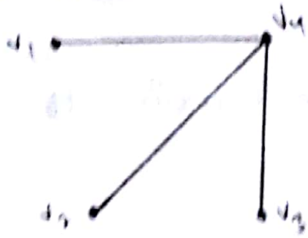
ii) $(v_1 - v_4)(v_2 - v_3)(v_1 - v_3)$



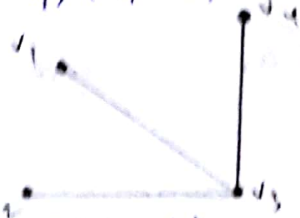
iii) $(v_2 - v_3)(v_2 - v_4)(v_2 - v_4)$



iv) $(v_1 - v_2)(v_2 - v_3)(v_1 - v_2)$



v) $(v_1 - v_3)(v_1 - v_4)(v_4 - v_2)$

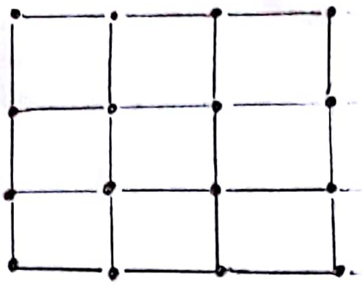


vi) $(v_1 - v_4)(v_4 - v_3)(v_3 - v_1)$

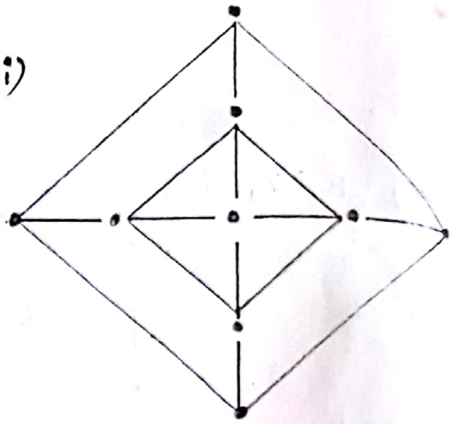
Scanned by CamScanner

6) Find a spanning tree for each of the graph shown by

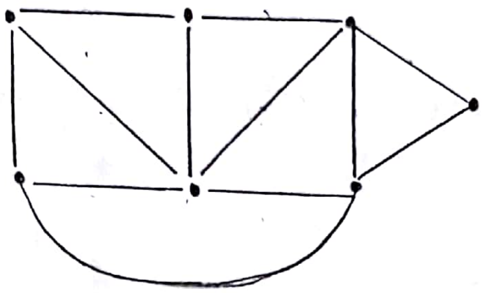
i)



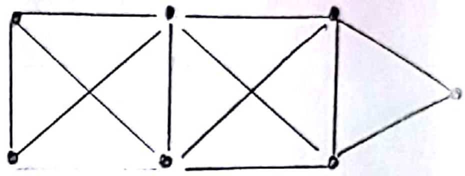
ii)



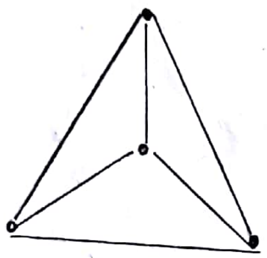
iii)



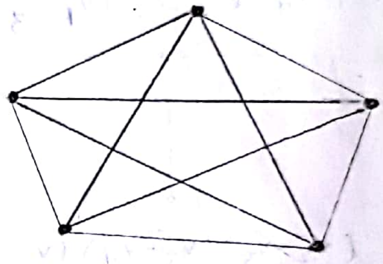
iv)



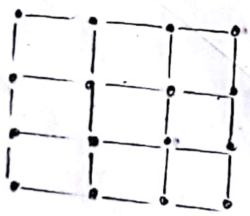
v)



vi)



i) Sol

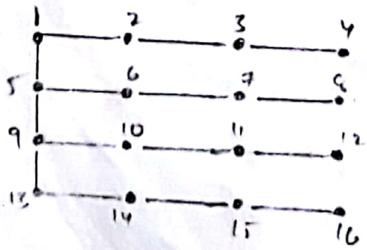


In the above graph we contain 16 vertices 24 Edges

$m-n+1$

$= 24-16+1$

$= 9$ edges are deleted



Delete (2-6) (3-7) (4-8)

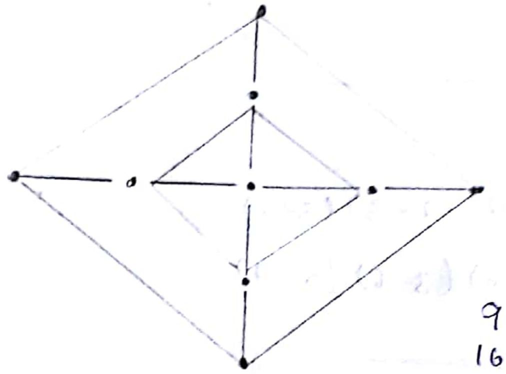
(6-10) (7-11) (8-12)

(10-14) (11-15) (12-14)

9 Edges are deleted and the above figure is required spanning tree

ii)

Sol



9 - V

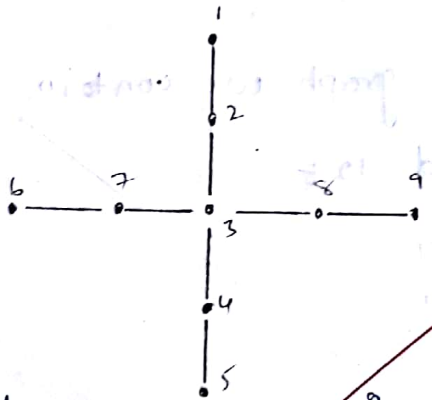
16 - E

$m - n + 1$

$= 16 - 9 + 1$

$= 7 + 1$

$= 8$



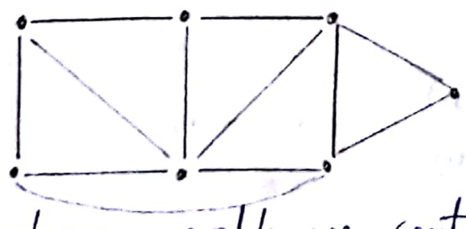
Deleted edges are 8 in the above graph

Delete (1,6) (6,5) (5,9) (9,1)

(2,7) (7,4) (4,8) (8,2)

iii)

Sol



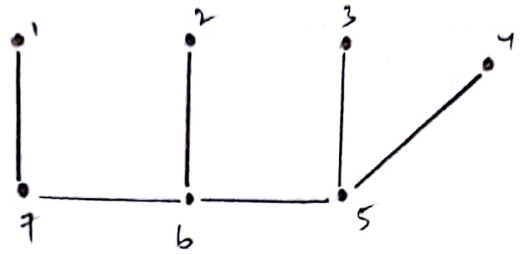
In above graph we contain 7 vertices and 12 Edges

$= m-n+1$

$= 12-7+1$

$= 5+1$

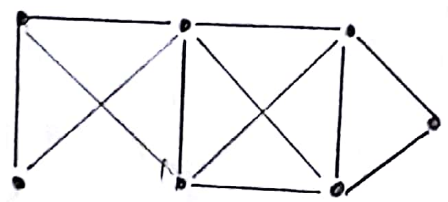
$= 6$



Delete (1-2) (2-3) (3-4)

(1-6) (3-6) (5-7)

iv)



In the above graph we contain

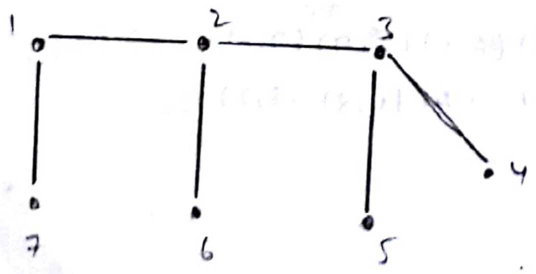
7V and 12E

$m-n+1$

$= 12-7+1$

$= 5+1$

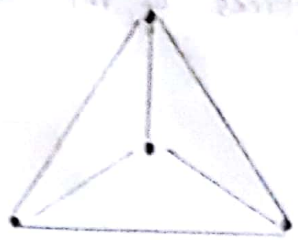
$= 6$



Delete (1-6) (2-7) (2-5)

(6-3) (6-5) (5-4)

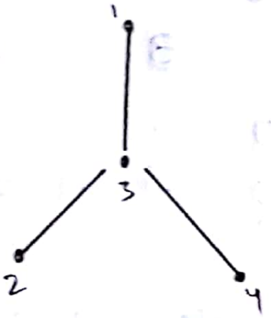
5)



In above graph we contain

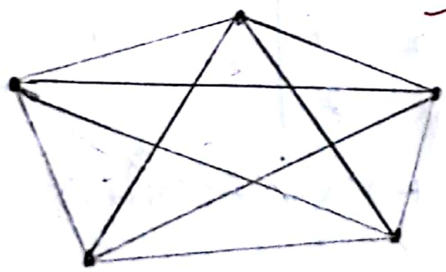
4 V
6 E

$m-n+1$
$= 6-4+1$
$= 2+1$
$= 3$



Delete
(1-2) (2-4) (4-1)

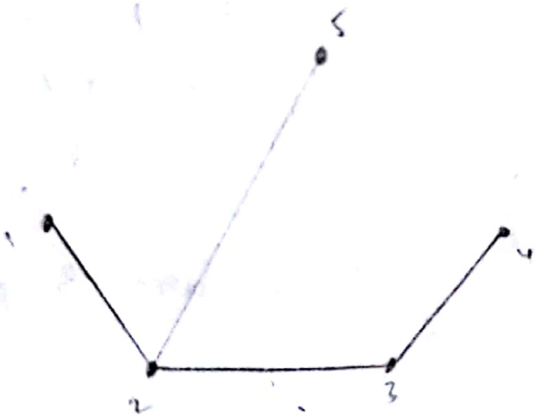
6)



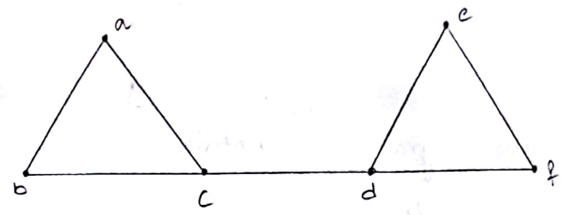
In above graph we contain

5 V
10 E

$= m-n+1$
$= 10-5+1$
$= 5+1$
$= 6$



7. Draw all spanning trees of the following graphs



Sol

$$m = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \end{array}$$

$m - n + 1$
$= 7 - 6 + 1$
$= 1 + 1$
$= 2$

The co-factor of the element $3(m_{33})$ is the determinent

$$\begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{vmatrix}$$

$= 3(4-1) - (-1)(-2-1) + (-1)(1+2)$

$= 3(3) + (-3) - (3)$

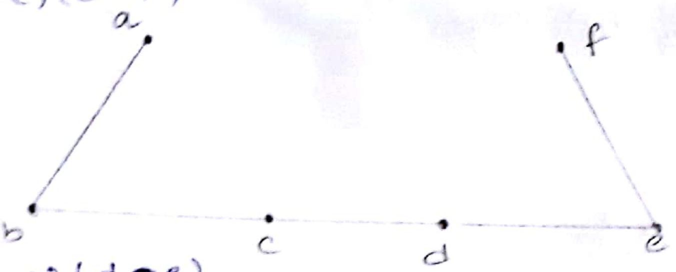
$= 9 - 3 - 3$

$= 9 - 6$

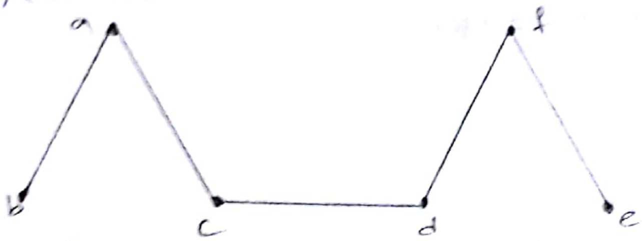
$= 3$

There are 3 spanning trees

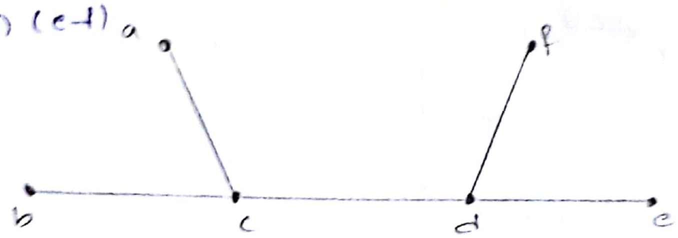
(i) (a-c)(d-f)



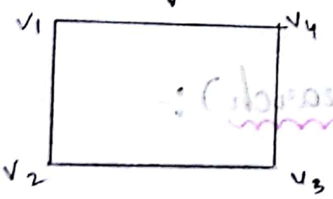
ii) (b-c)(d-e)



iii) (a-b)(e-f)



8. Draw all spanning trees of the following graph



m = edge
n = vertices

Sol

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix} \end{array}$$

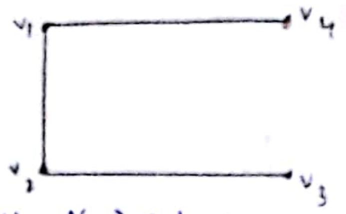
$= m - n + 1$
$= 4 - 4 + 1$
$= 1$

The co-factor of the element $2 (m_{11})$ is the determinent

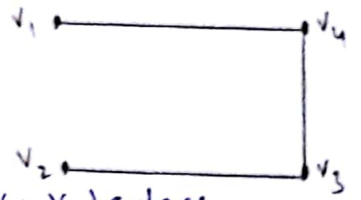
$$D = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{vmatrix}$$

$= 2(4-1) - (-1)(-2-0)$
$= 2(3) + 1(-2)$
$= 6 - 2 = 4$

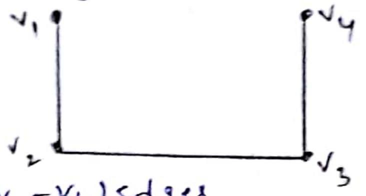
i) Delete $(v_3 - v_4)$ Edges



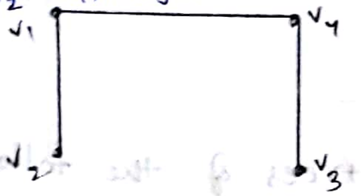
ii) Delete $(v_1 - v_2)$ Edges



iii) Delete $(v_1 - v_3)$ edges



iv) Delete $(v_2 - v_4)$ Edges



BFS (Breadth first search) :-

Algorithm :-

Step: 1 Orbitarly choose a vertex and consider it is a root. Then add all edges incident to this vertex, such that the addition of edges does not form any cycle

Step: 2 The new vertices added at this stage become the vertices at level, one in a shanning tree, orbitarly order them

Step: 3 Next, for each vertex at level-1, visited in order, and add each edge incident to this vertex to the tree as long as it does not form any cycle

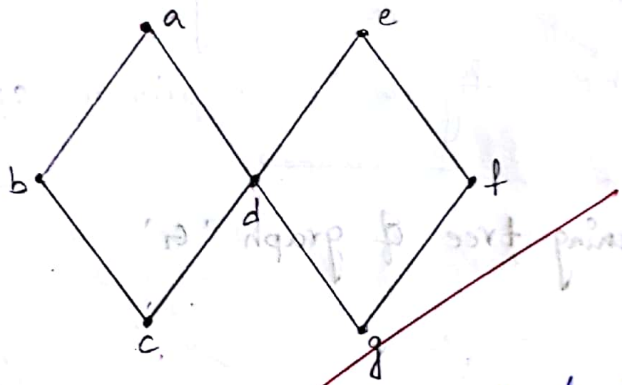
Step:4 Orbitarly order the children of each vertex at level-1 this processor, the vertices at level-2 in the tree.

Step: 5 Continue the same procedure until all the vertices in the tree have been added.

Step: 6 The procedure ends, since there are only finite number of edges in the graph.

Step: 7 A spanning is produced since we have produced a tree with out cycle containing every vertex of the graph.

1. Eg:- Find spanning tree of graph 'G' by using BFS Algorithm

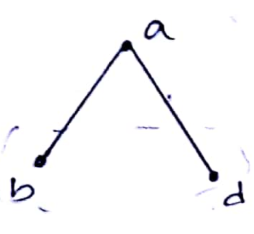


Sol

Step 1:- choose the vertex that to be a root.

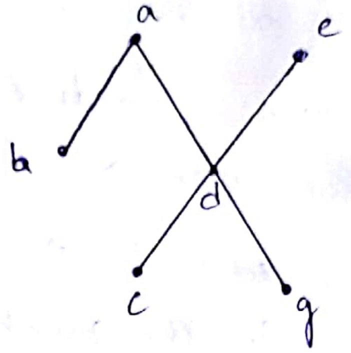
a. (vertex a)

Step 2:- Add edges incident with all vertices adjecent to 'a'. So, that edges {a, b}, {a, d} added



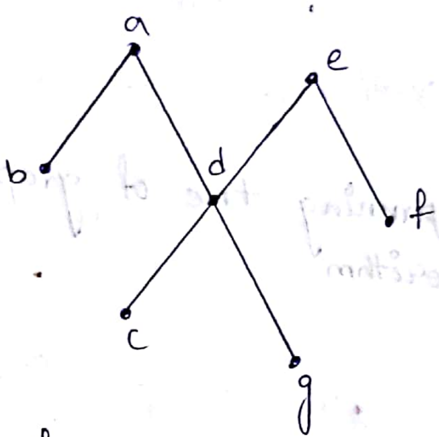
The two vertices b, d are in level-1 in the tree

Step 3:- Add edges from this vertices at level-1 to adjecent vertices



b, c edge will form a cycle so it is deleted

Step 4:- Add edge in level-2 to adjecent vertices.

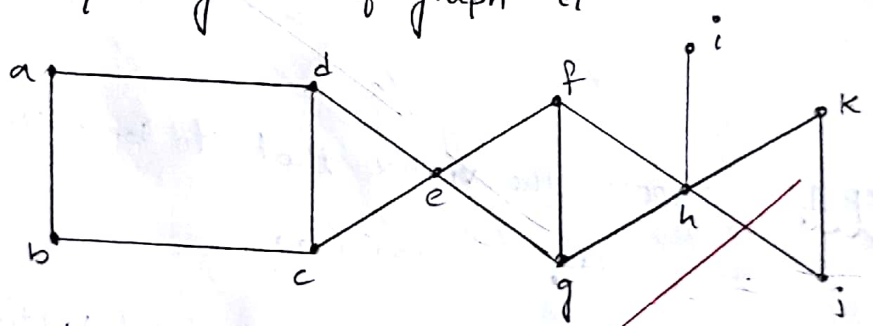


g, f edges will form a cycle so it is deleted

The above figure is required spanning tree.

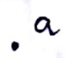
14M, SMP

2. Find spanning tree of graph 'G'

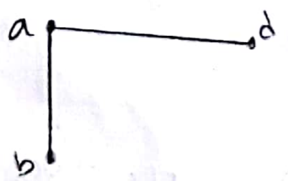


Sol

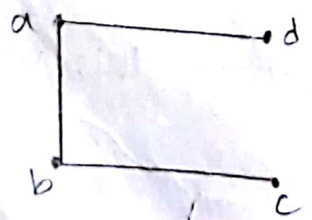
Step 1:-



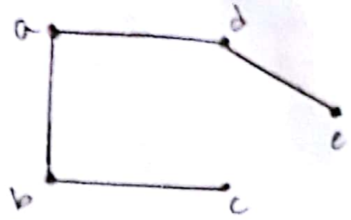
Step 2:-



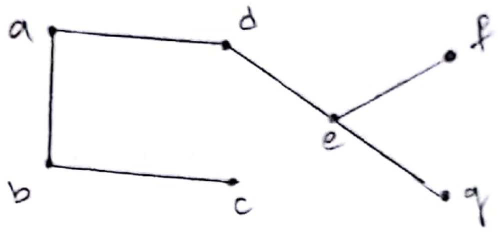
Step 3:-



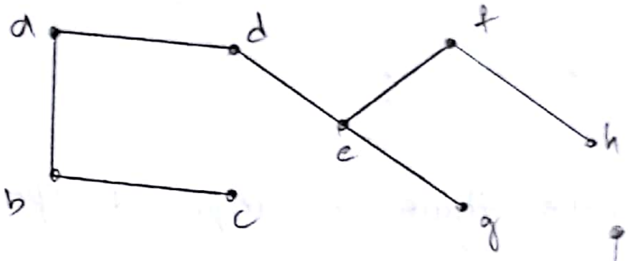
Step 4:-



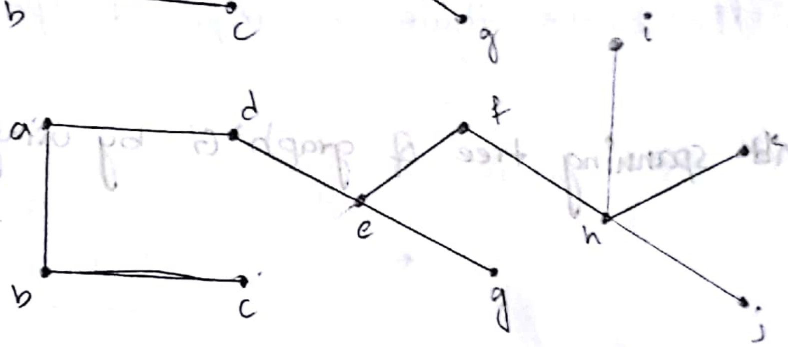
Step 5:-



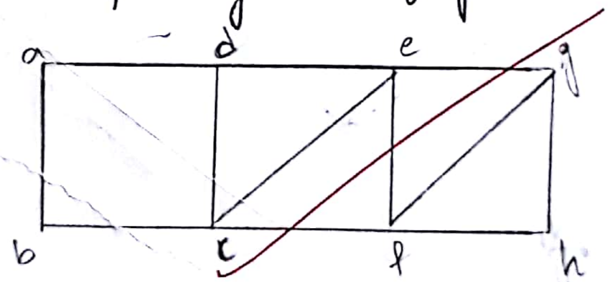
Step 6:-



Step 7:-

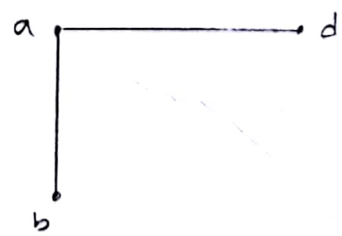


3. Find the spanning tree of graph 'G' by using BFS

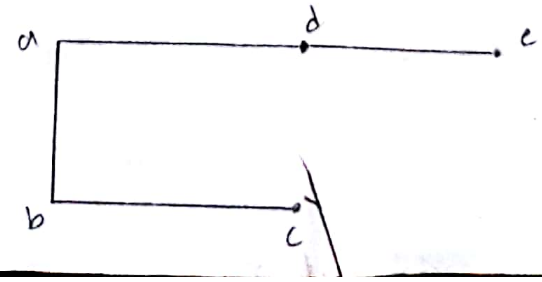


Step 1:-

Step 2:-

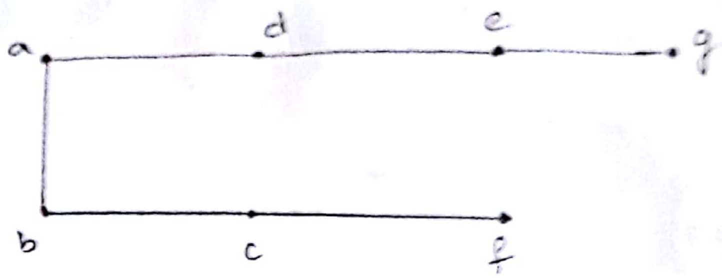


Step 3:-

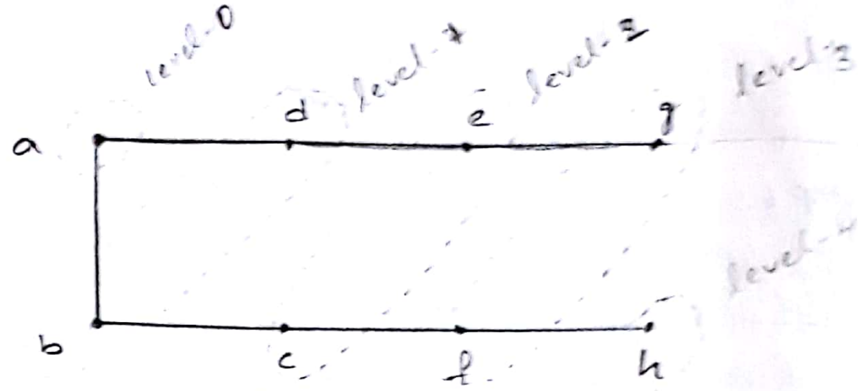


Scanned by CamScanner

Step 4:-

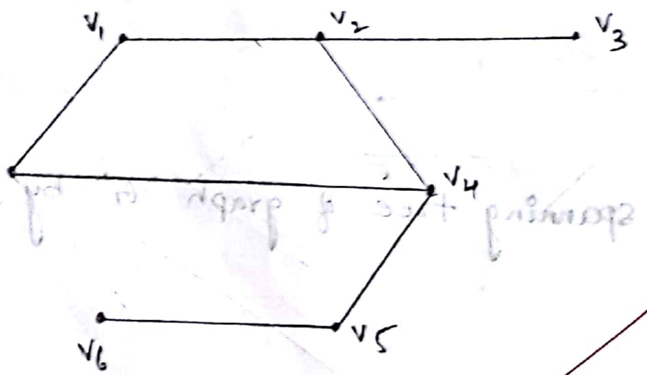


Step 5:-



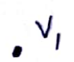
The above figure is required spanning tree

4. find spanning tree of graph 'G' by using BFS Algorithm

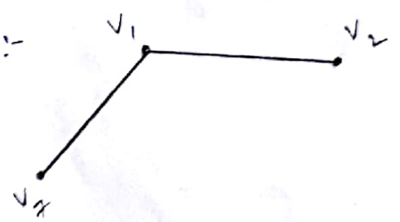


Sol

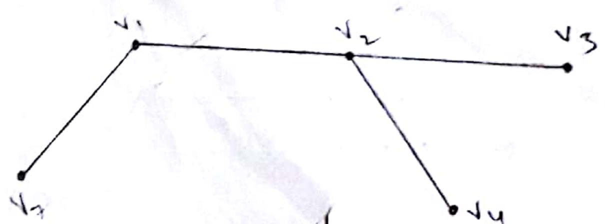
Step 1:-



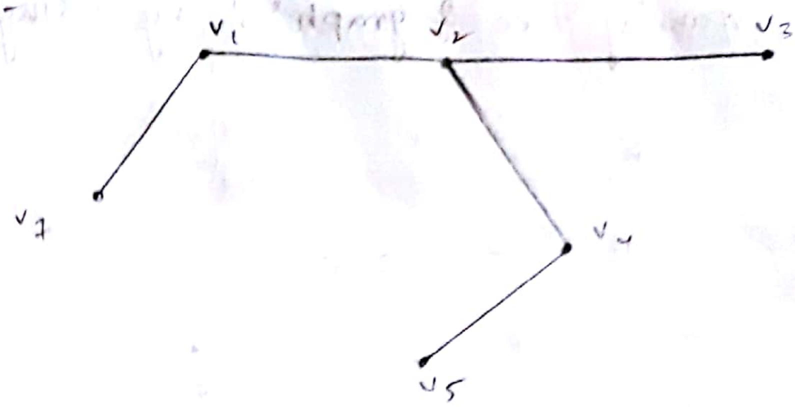
Step 2:-



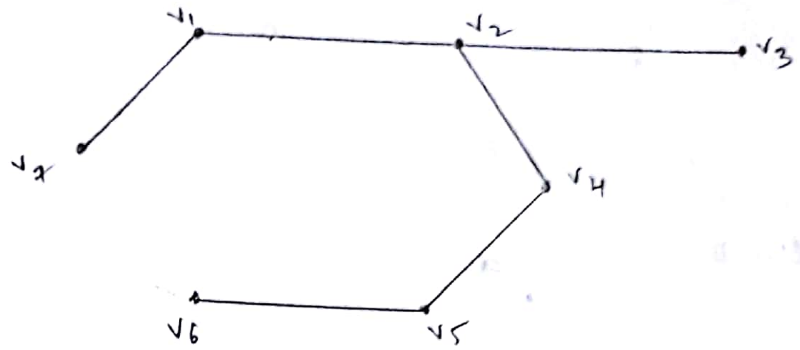
Step 3:-



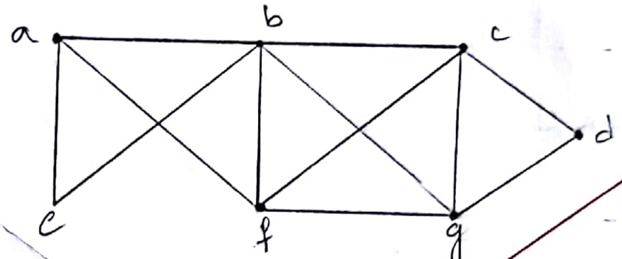
Step 4:-



Step 5:-

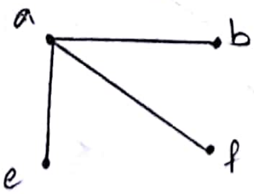


5. find spanning tree of graph 'G' by BFS

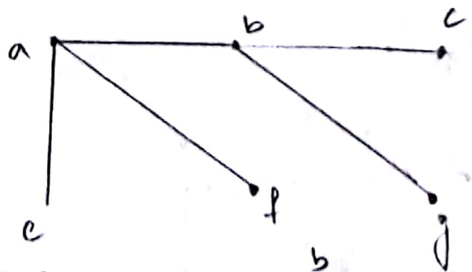


Sol:- Step:1:-

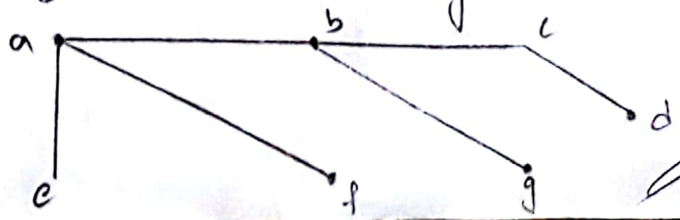
Step: 2:-



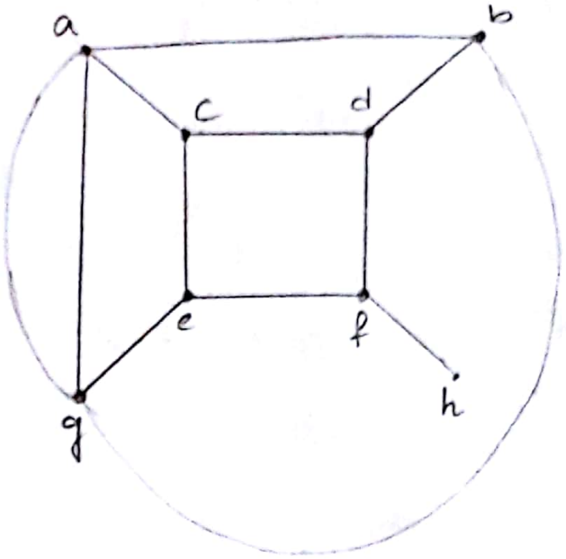
Step: 3



step 4:-

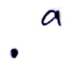


6. find spanning tree of graph 'G' by using BFS

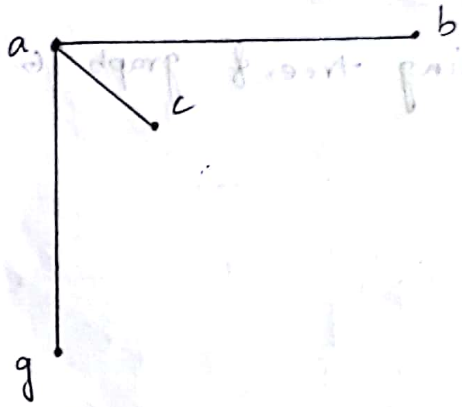


Sol

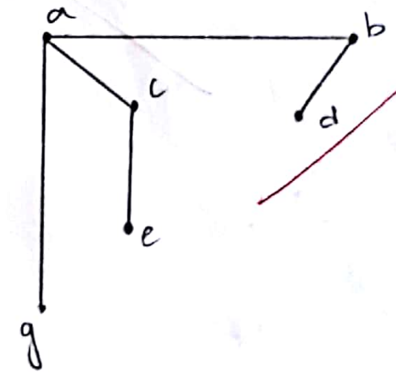
Step 1 :-



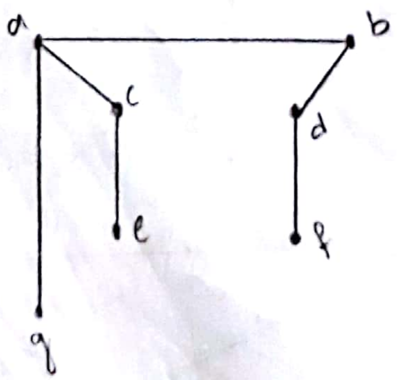
Step 2 :-



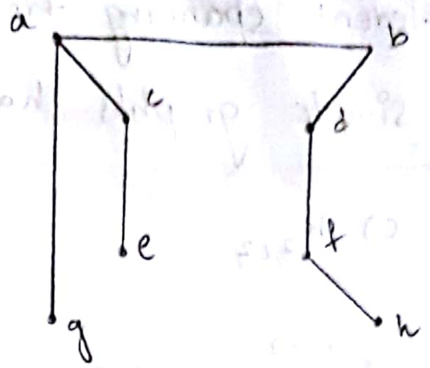
Step 3 :-



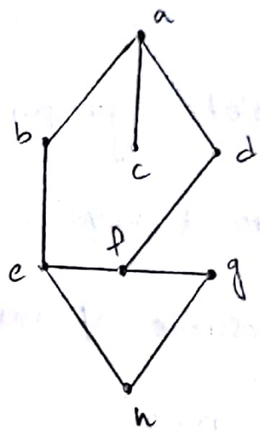
Step 4 :-



Step 5 :-



7. find spanning tree of graph 'G' by using BFS

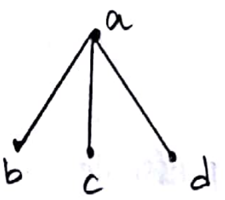


Sol

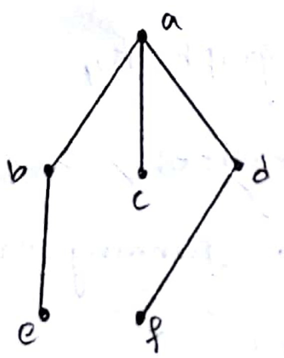
Step 1 :- • a

Indrani Sutar

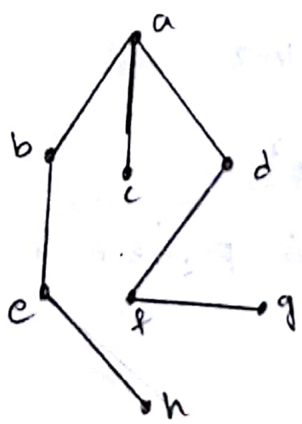
Step 2 :-



Step 3 :-



Step 4 :-



8. How many different spanning trees does each of the following simple graph's have

a) K_3 b) K_4 c) $K_{2,2}$

Note :- for K_n , n^{n-2} different possibile ways

$K_{n,n}$, n^{2n-2} different possibile ways

Sol

a) Given complete graph K_3

where 3 = no. of vertices

$\therefore$ The possible spanning trees

$$K_n = n^{n-2}$$

$$K_3 = 3^{3-2}$$

$$= 3^1$$

$K_3 = 3$ possible ways

b) Given complete graph K_4

4 = no. of vertices

$\therefore$ The possible spanning trees

$$K_n = n^{n-2}$$

$$K_4 = 4^{4-2}$$

$$= 4^2$$

$K_4 = 16$ possible ways

c) Given complete graph $K_{2,2}$

Sol:

∴ The possible spanning trees

$K_{2,n} = n^{2n-2}$

$K_{2,2} = 2^{2(2)-2}$

$= 2^{4-2}$

$= 2^2$

$K_{2,2} = 4$ possible ways

DFS (Depth first search) :-

An alternative to breath first search is depth first search which proceeds to successive levels in a tree at ~~area~~ earliest possible oppurtunity. DFS is also called "Back Tracking".

Algorithm:-

Step: 1 orbitoarily choose a vertex from the vertices of the graph and it is a root.

Step: 2 From the path starting at this vertex by successfully adding edges as long as possible where each new edge is incident with the last vertex in a path ~~wt~~ without producing any cycle

Step: 3 If the path goes through all vertices of the graph the tree consisting of the path is a spanning tree otherwise move back to the next to last vertex in the path, if possible, from a new path starting at this vertex passing through vertices that were ~~are~~ not already visited.

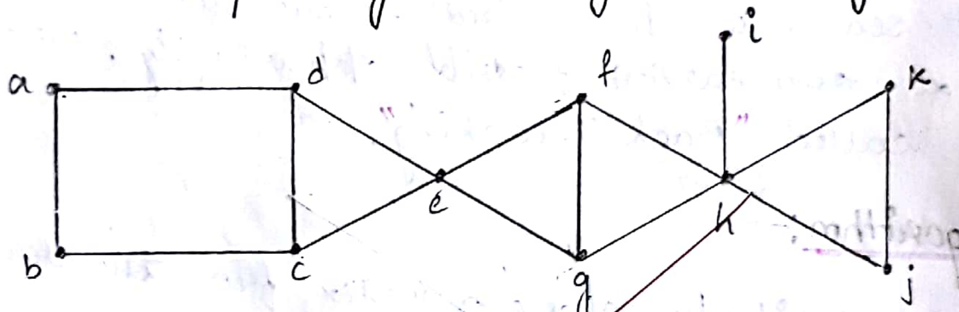
Step 4: if the cannot be done, move back another vertex in the path, that is ~~two~~ to vertices back in path.

step 5: Repeat this procedure, begining at the last vertex visited, moving backup the path one vertex at a time, forming new paths that are as long as possible untill no more edges can bee added

Step 6: This process ends since the graph has a finite no.of edges and is connected there fore a Spanning tree is produced.

DFS (Depth first search) :-

1. Eg:- Find a spanning tree of graph 'G' by using DFS



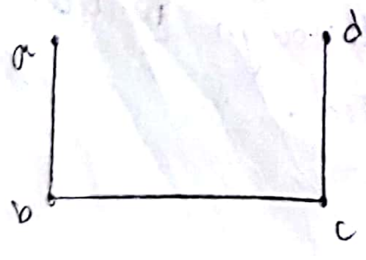
Sol

① a

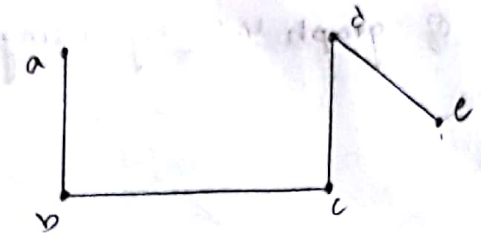
②

③.

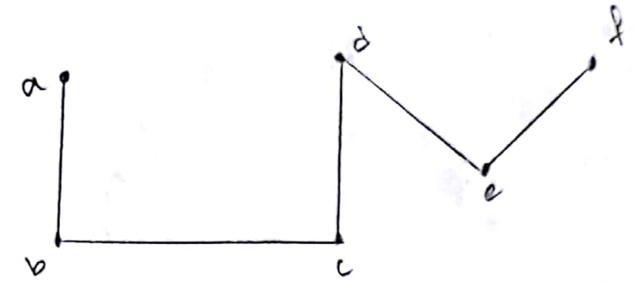
④



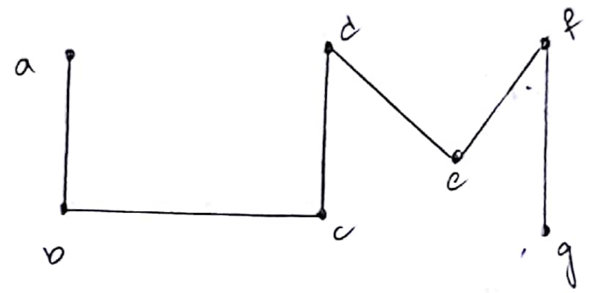
5


a
b
c
d
e

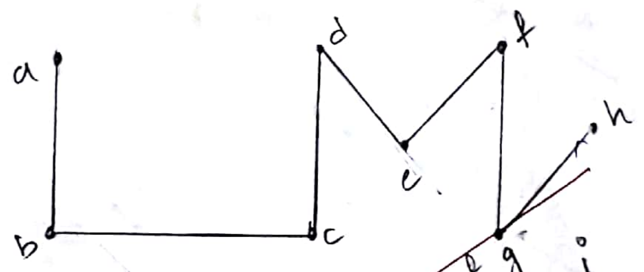
6


a
b
c
d
e
f

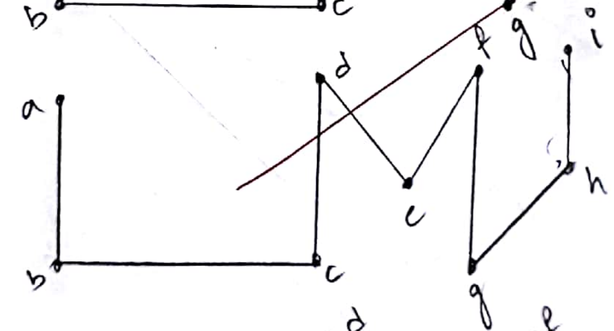
7


a
b
c
d
e
f
g

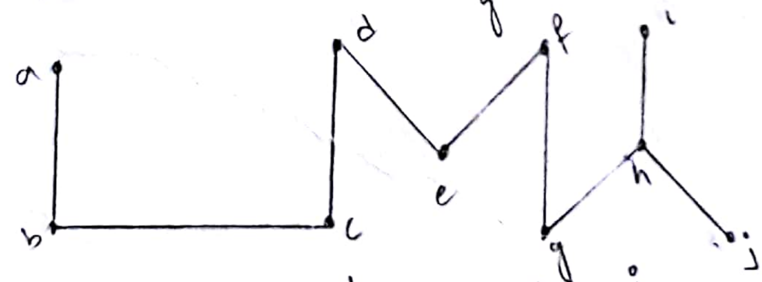
8


a
b
c
d
e
f
g
h

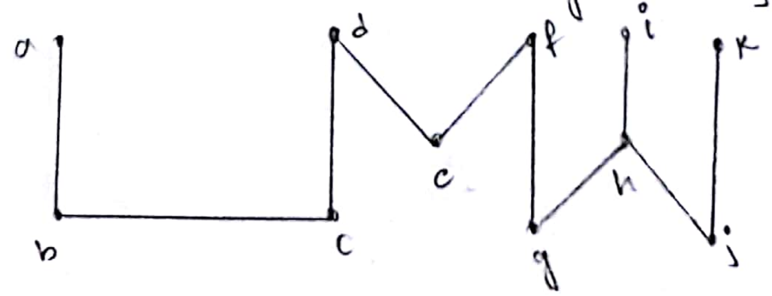
9


a
b
c
d
e
f
g
h
i

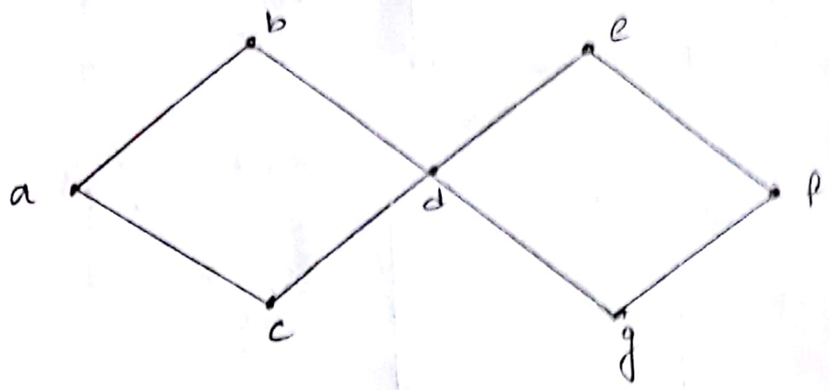
10


a
b
c
d
e
f
g
h
i
j

11

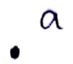

a
b
c
d
e
f
g
h
i
j
K

2. Find spanning tree of graph 'G' by using DFS Algori[thm]



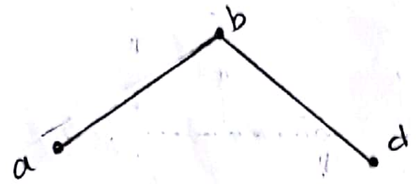
Sol

(1)

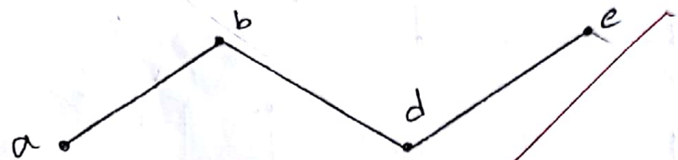


(2)

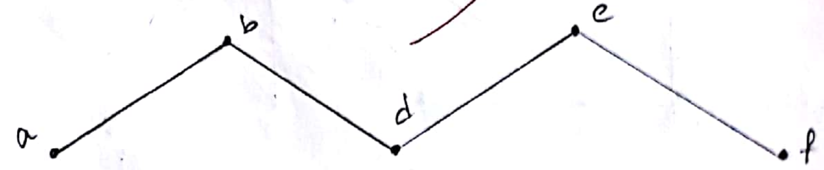
(3)



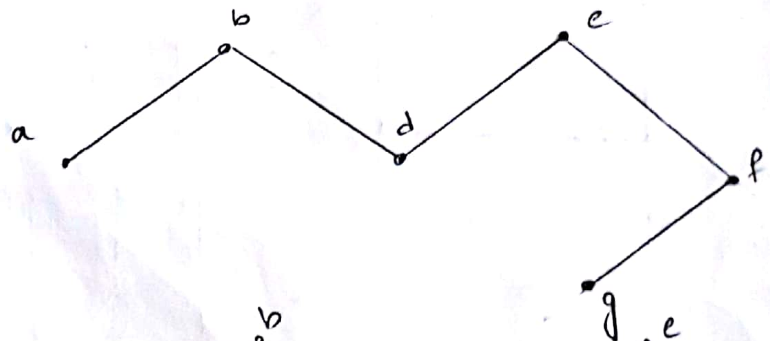
(4)



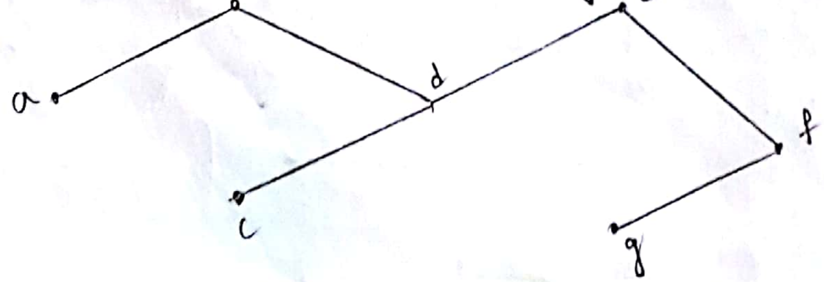
(5)



(6)



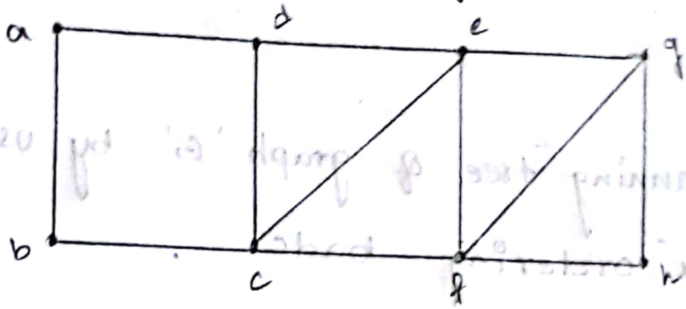
(7)



Explanation for 2nd problem:-

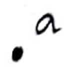
Choose a vertex 'a'. From a path by succefully adding edges incident with vertices not already in the path as long as possible. This produces the path a-b-d-e-f-g. Now back track to g. There is no path similary after back trak f and e here also there is no path. So move back track at 'd' from the path 'd-c'. This produces the required spanning tree by using DFS Algorithm

3) Find spanning tree of graph 'G' by using DFS.



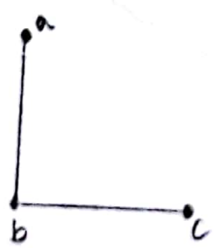
Sol

①

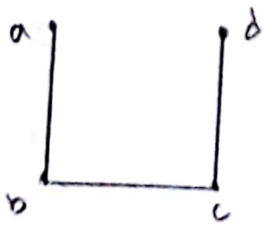


②

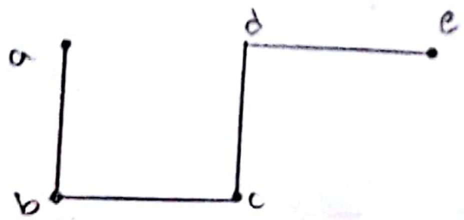
③



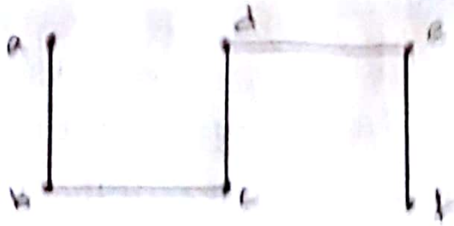
④



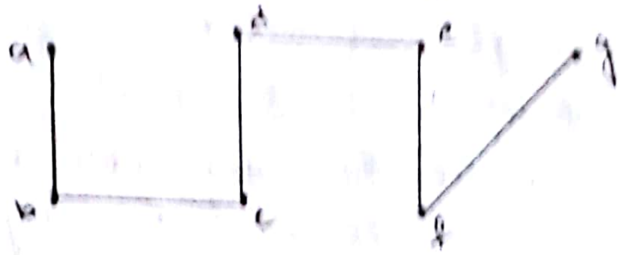
⑤



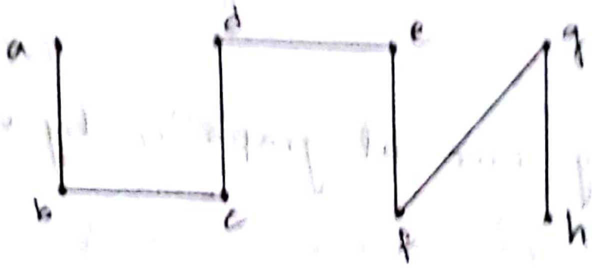
(6)



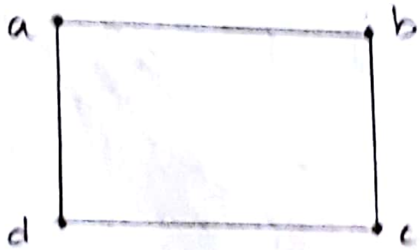
(7)



(8)



4. Find spanning tree of graph 'G' by using DFS with vertex ordering badc

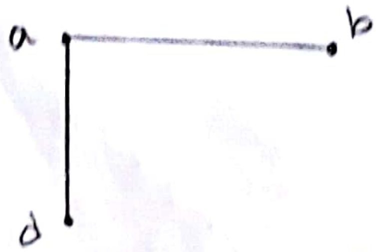


Sol

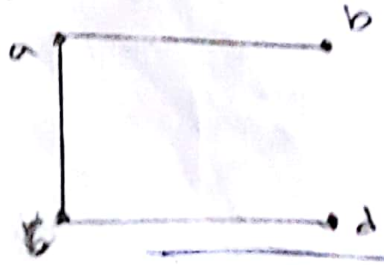
(1)

(2)

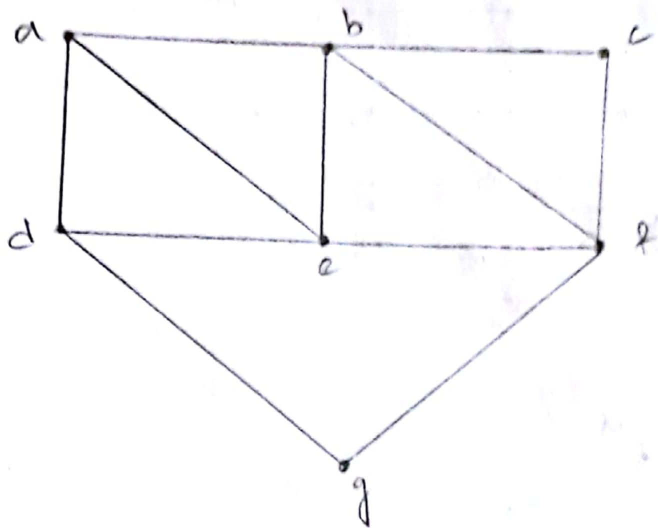
(3)



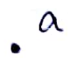
(4)



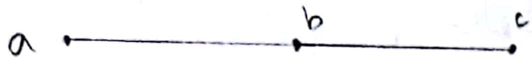
5) find spanning tree by of graph 'G' by using DFS with vertex ordering abcfedg

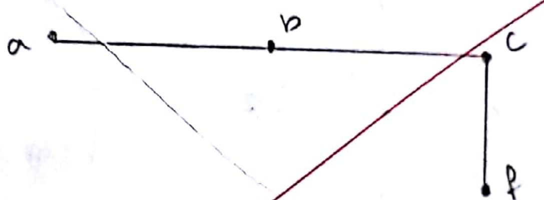


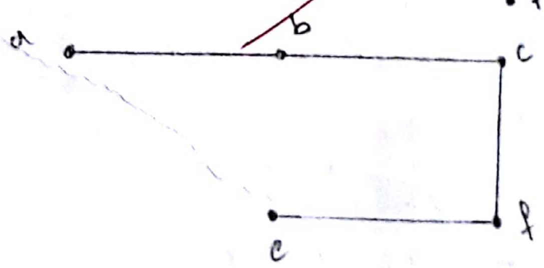
Sol

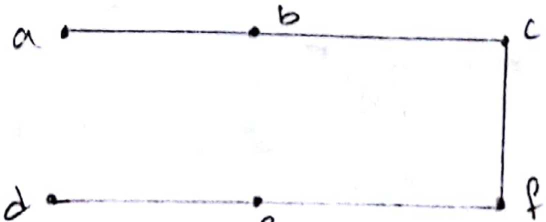
① 

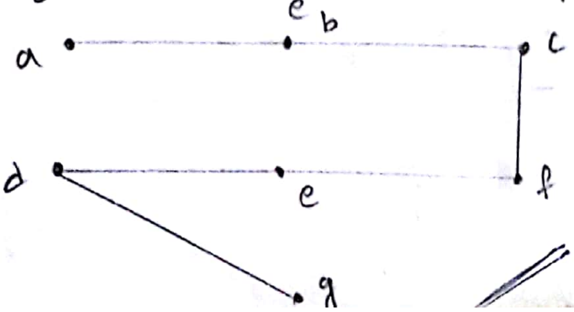
② 

③ 

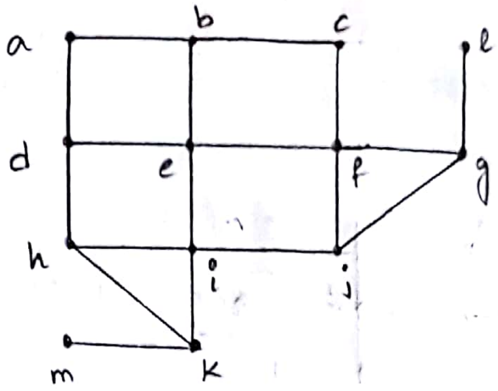
④ 

⑤ 

⑥ 

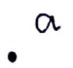
⑦ 

(6) Find a spanning tree of the graph 'G' by using DFS with vertex in the ordering abcfjihdegl



Sol:-

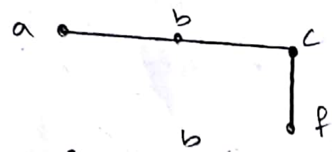
(1)



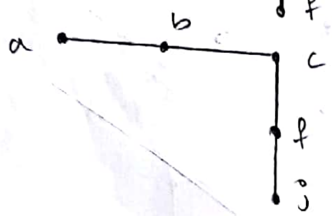
(2)

(3)

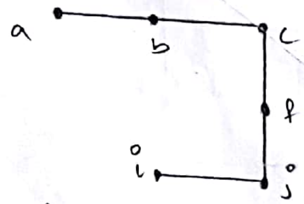
(4)



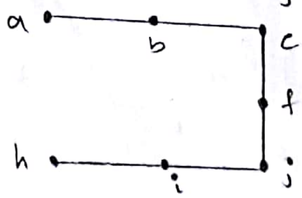
(5)



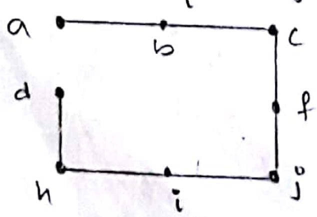
(6)



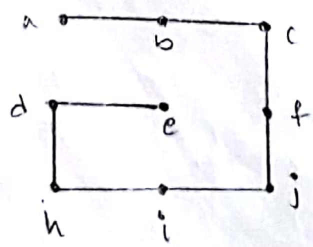
(7)



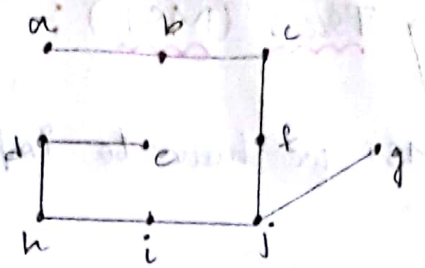
(8)



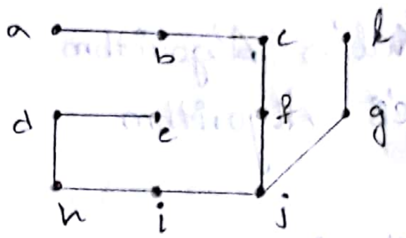
(9)



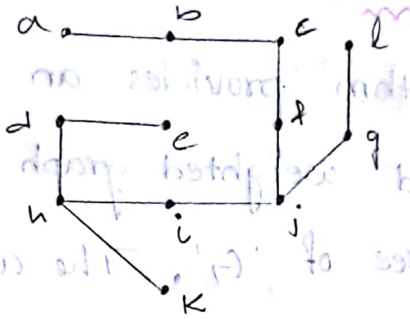
(10)



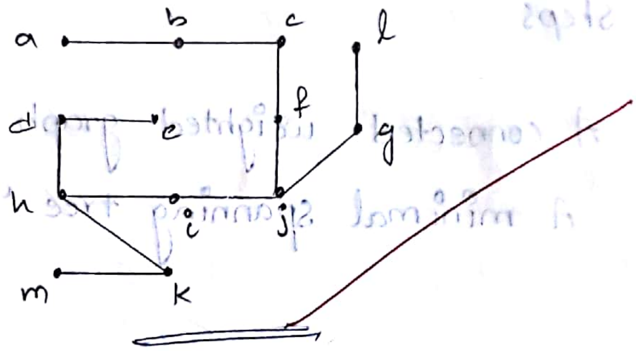
(11)



(12)



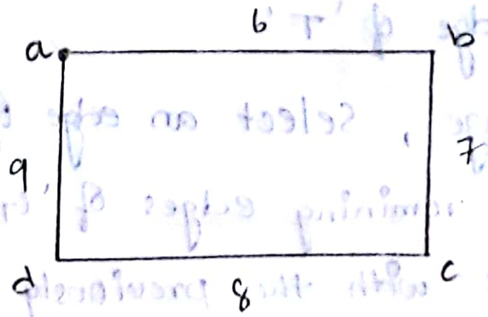
(13)



Weighted graph :-

A graph 'G' is said to be a weighted graph in which each edge has been assigned non-negative number is called a weighted graph. It can be represented as '$W(e)$'.

Eg:-



Minimal spanning Trees (MST) :-

In this MST we have to impliment two algorithms

1. Kruskal's Algorithm
2. prim's Algorithm.

Kruskal's Algorithm :-

This Algorithm provides an acyclic sub graph 'T' of a connected weighted graph 'G' which is a minimal spanning tree of 'G'. The algorithm involves following steps

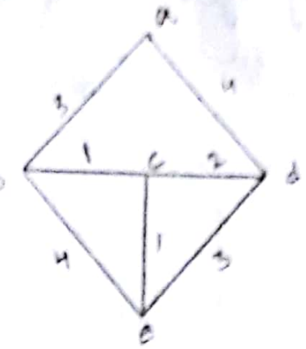
input :- A connected weighted graph 'G'

output :- A minimal spanning tree 'T' of 'G'

Steps :-

* ① List all the edges (which do not form loop) of 'G' in non decreasing order of the weights
* ② Select an edge of minimum weight (if more than one edge of minimum weight orbitarly choose one of them). This is the 1st edge of 'T'
* ③ At each stage, select an edge of minimum weight of all the remining edges of 'G' and it does not form any cycle with the previously selected edges in T. Add the edge to 'T'
* ④ Repeat step ③ untill (n-1) edges have been selected where n is no. of vertices.

1. Find minimal spanning tree by using kruskal's Algorithm

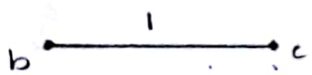


Sol

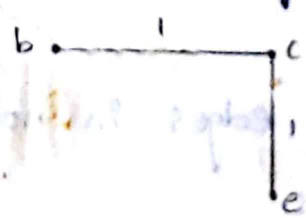
① List the all edges in non-decreasing order of their weights

Edge	Weight
(b, c)	1
(c, e)	1
(c, d)	2
(d, e)	3
(b, a)	3
(b, e)	4
(a, d)	4

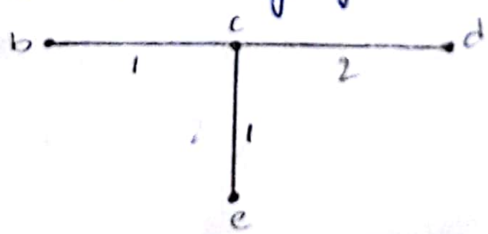
② select the edge (b, c) since it has the smallest weight including it in T.



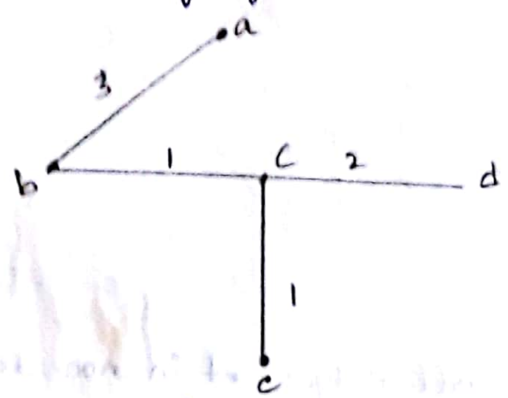
③ select an edge with Next smallest weight (c, e) since it does not form any cycle with existing edges



④ select an edge with Next smallest weight (c, d) Since it does not form any cycle with existing edges



(5) Select an edge with next smallest edge (a, b) since it does not form any cycle with the existing edges.

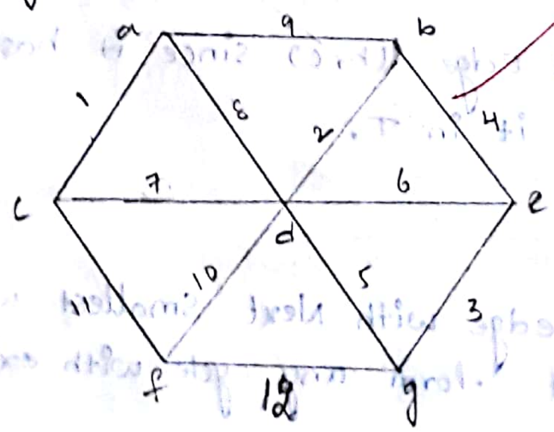


(6) Stop the procedure in the above graph contain 5 vertices and 4 edges.

Hence the required minimal spanning tree is

Total cost = 3+2+1+1
= 7

2. Find minimal spanning tree of graph 'G' by using kruskal's algorithm.

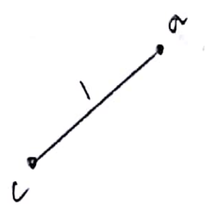


(8) Sol

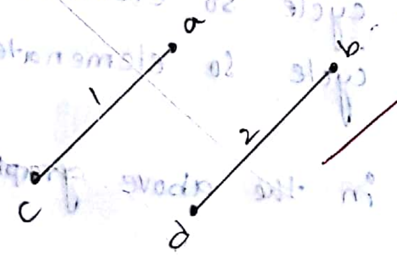
(1) List all the edges in non-decreasing order of their weights.

Edge	weight
(a,c)	1
(b,d)	2
(e,g)	3
(b-e)	4
(d-g)	5
(d-e)	6
(c-d)	7
(a-d)	8
(a-b)	9
(d-f)	10
(c-f)	11
(f-g)	12

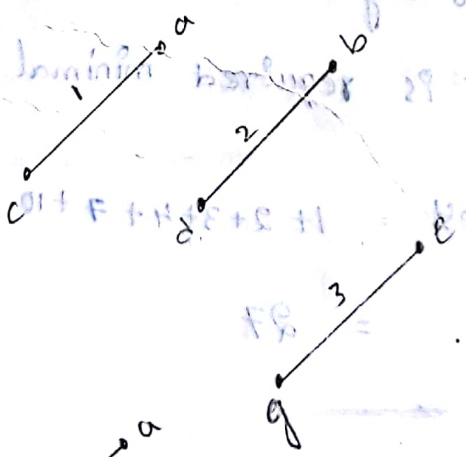
(2)



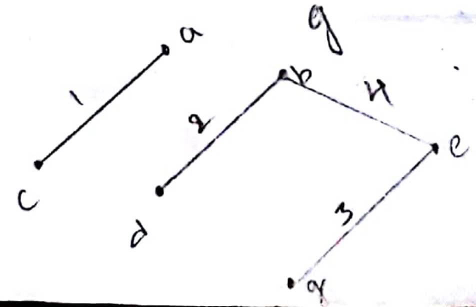
(3)



(4)



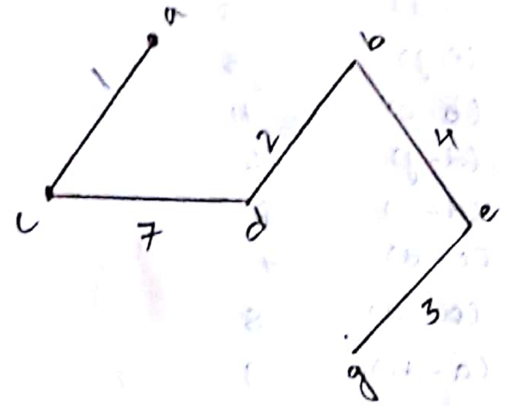
(5)



⑥ (d,g) edges form a cycle so elemenate it.

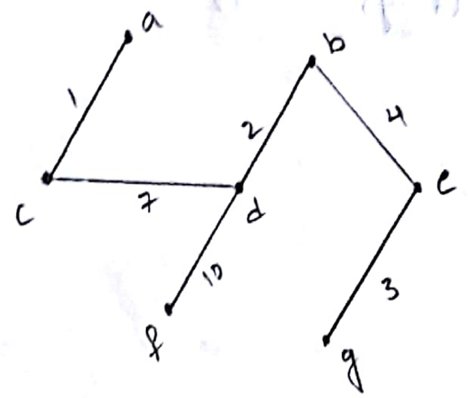
⑦ (d,e) edges form a cycle so elemenate it.

⑧



⑨ (a,b) edge form a cycle so elemenate it.

⑩



⑪ (c-f) edge form a cycle so elemenate it.

⑫ (f-g) edge form a cycle so elemenate it.

⑬ Stop the procedure in the above graph contain 7 vertices and 6 edges

Hence it is required minimal spanning tree

Total cost = 1+2+3+4+7+10

= 27

Prim's Algorithm :-

Initially algorithm starting at a disignated vertex chooses an edge with minimum weight and consider this edge and associated vertices as part of the desired tree. In this process terminate when (n-1) edges have been selected from a graph 'G'. The Algorithm involves the following steps

Input :- A connected weighted graph 'G'

Out put :- A minimal spanning tree 'T' of 'G'

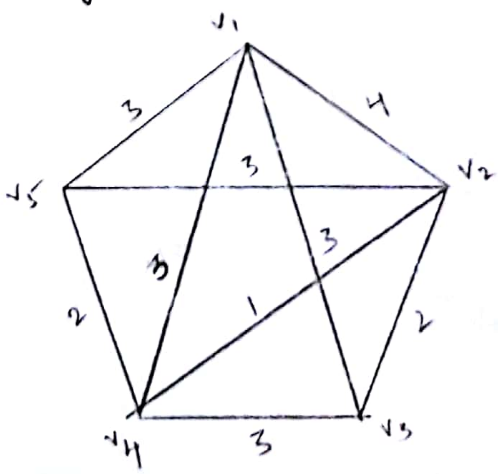
Steps :-

① Select any vertex in 'G'. Among all the edges incident with selected vertices, choose an edge with minimum weight.

② At each stage, choose an edge of smallest weight joining a vertex already included in tree and a vertex not yet include and it doesnot form any cycle.

③ Repeat untill all vertices of 'G' included and (n-1) edges has been selected.

① find minimal spanning tree of weighted graph 'G' by using prim's algorithm.



Sol

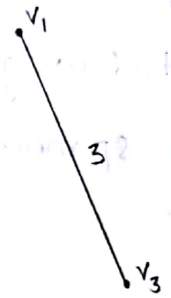
① We choose the vertex v_1, Now edge with Smallest incident on v_1 is

$(v_1, v_2) = 4$

$(v_1, v_3) = 3$

$(v_1, v_4) = 3$

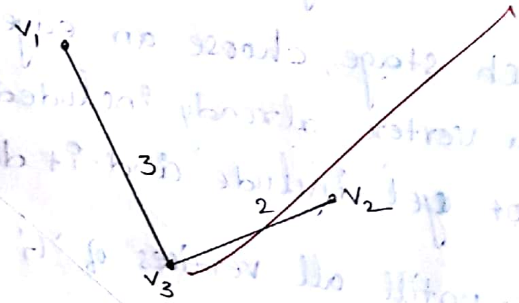
$(v_1, v_5) = 3$



② Now Smallest weight incident on v_3

$w(v_3, v_2) = 2$

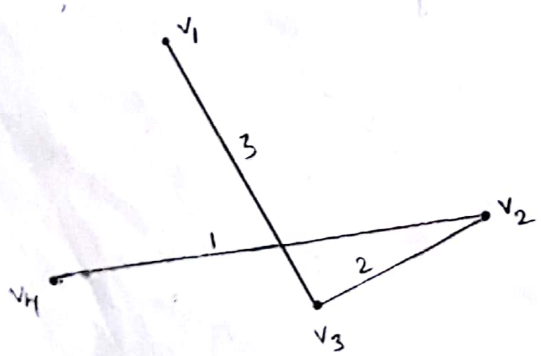
$w(v_3, v_4) = 3$



③ Now edge with Smallest weight incident on v_2

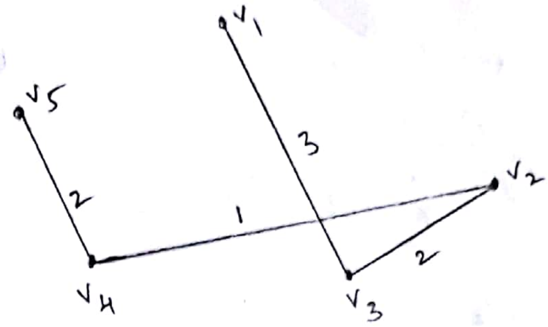
$w(v_2, v_4) = 1$

$w(v_2, v_5) = 3$



(4) Now edge with smallest weight incident on v_4

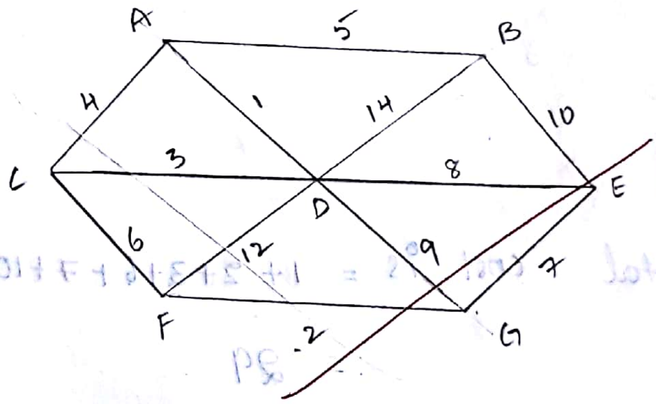
$w(v_4, v_5) = 2$



Total cost is $1+2+2+3$

$= 8$

2) find minimal spanning tree by using prim's algorithm for the given weighted graph.

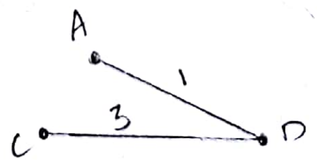


Sol

(1)

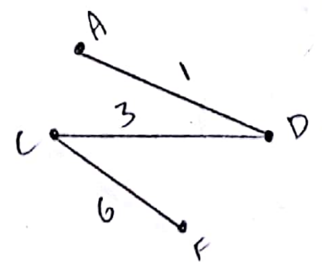
$w(A,C) = 4$
$w(A,B) = 5$
$w(A,D) = 1$

(2)



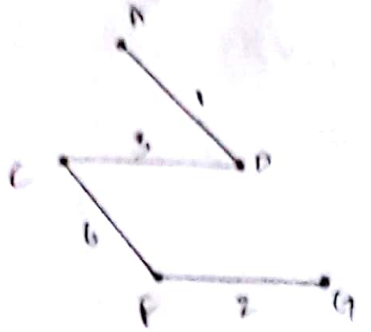
$w(D,C) = 3$

(3)



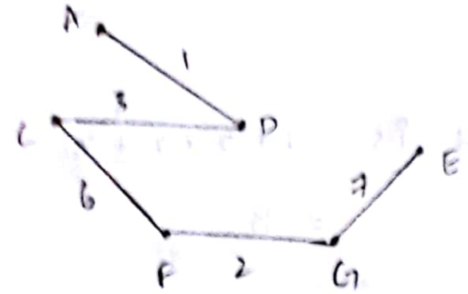
$w(C,F) = 6$

(4)



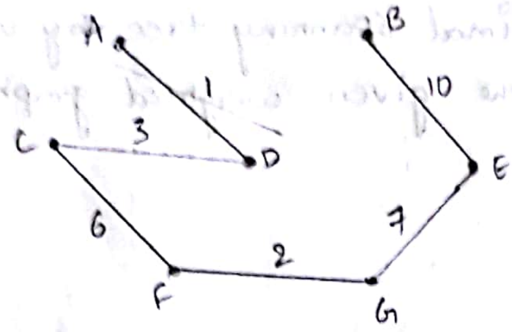
$w(F,G) = 2$

(5)



$w(G,E) = 7$

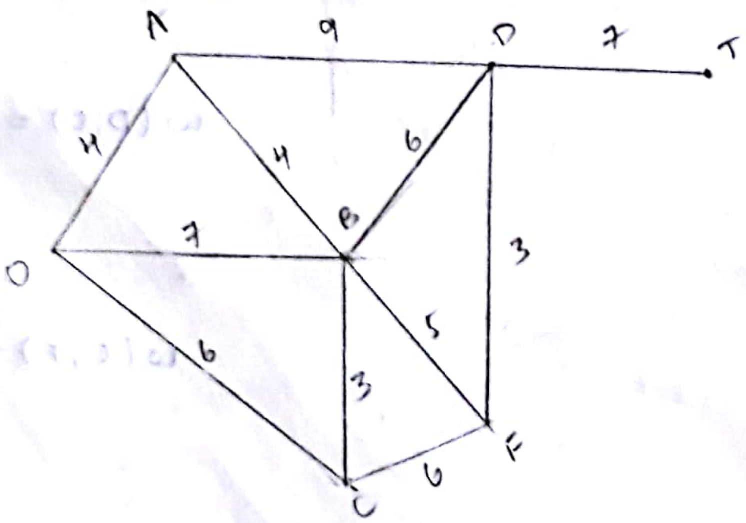
(6)



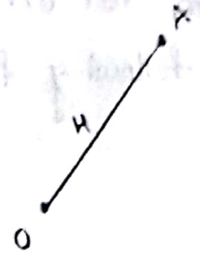
$w(E,B) = 10$

Total cost is $= 1+2+3+6+7+10$

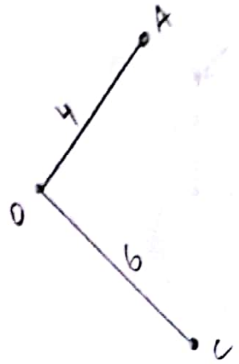
$= 29$



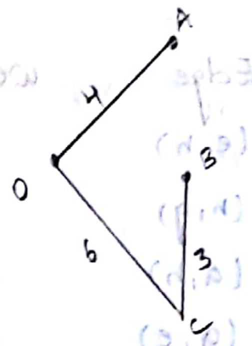
1



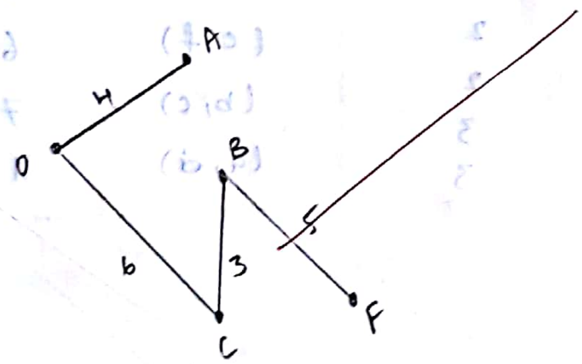
2



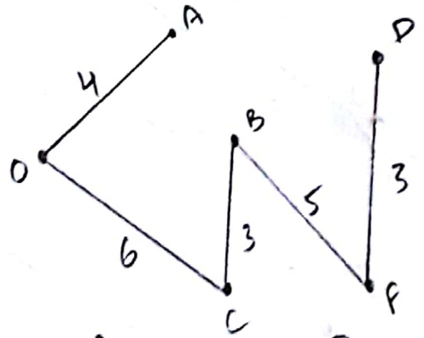
3



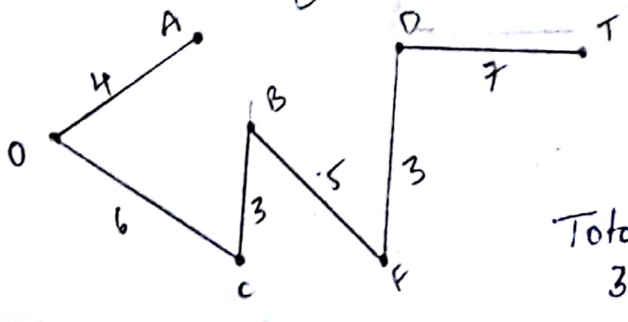
4



5



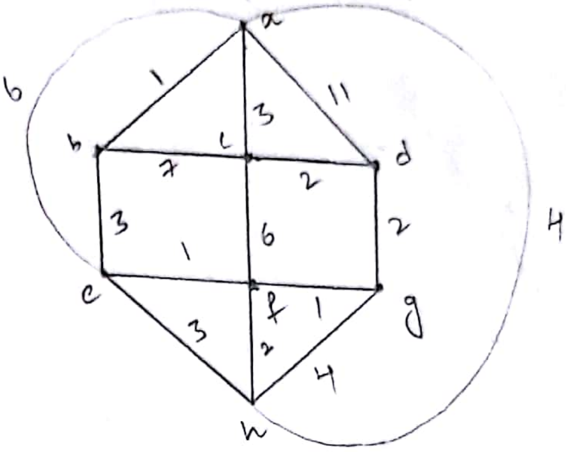
6



Total cost is =

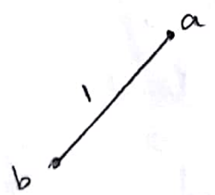
$3+3+4+5+6+7 = 28$

4. Explain kruskal's algorithm for orbitoaring the minimal spanning tree for the following figure.

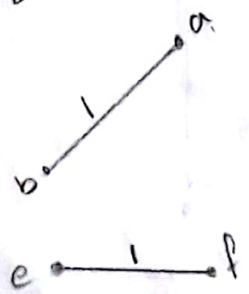


Edge	weight	Edge	weight
(a,b)	1	(e,h)	3
(e,f)	1	(h,g)	4
(f,g)	1	(a,h)	4
(c,d)	2	(a,e)	6
(d,g)	2	(c,f)	6
(f,h)	2	(b,c)	7
(a,c)	3	(a,d)	11
(b,e)	3		

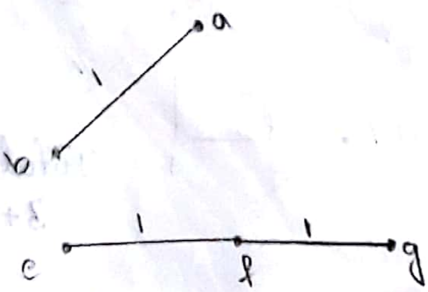
(1)



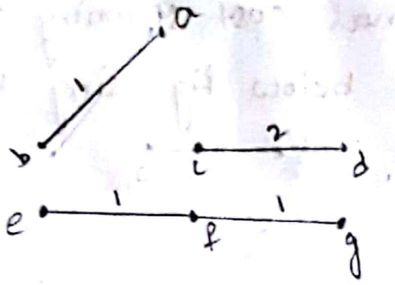
(2)



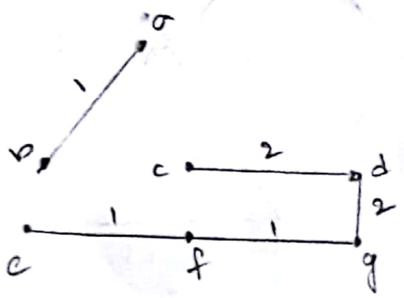
(3)



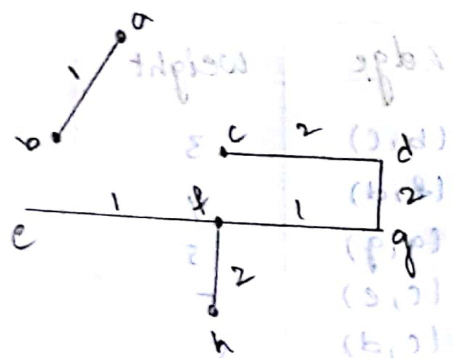
(4)



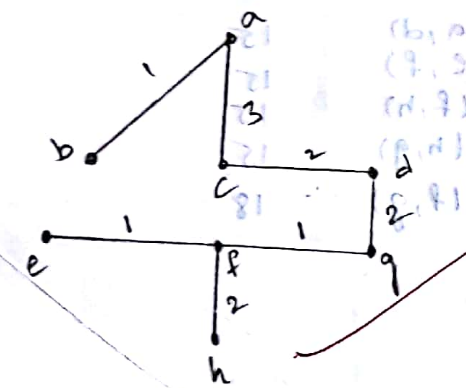
(5)



(6)



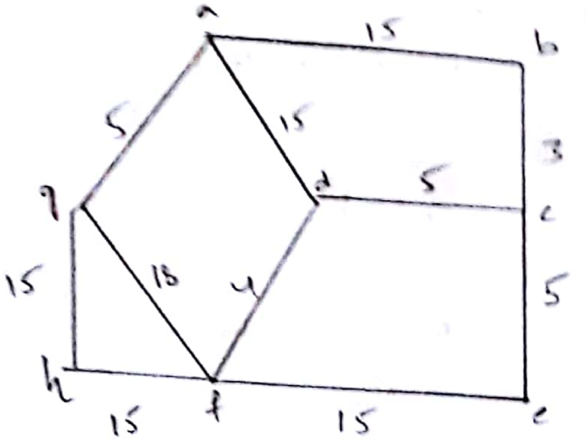
(7)



Total cost is $= 1+1+1+2+2+2+3$

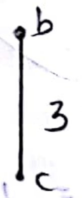
$= 12$

5. Construct minimal cost spanning tree for the cities shown in below fig using krushkal's algorithm

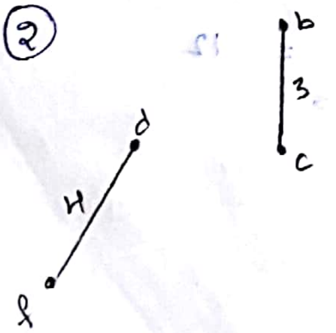


Edge	weight
(b,c)	3
(f,d)	4
(a,g)	5
(c,e)	5
(c,d)	5
(a,b)	15
(a,d)	15 ×
(e,f)	15 ×
(f,h)	15
(h,g)	15 ×
(f,g)	18 ×

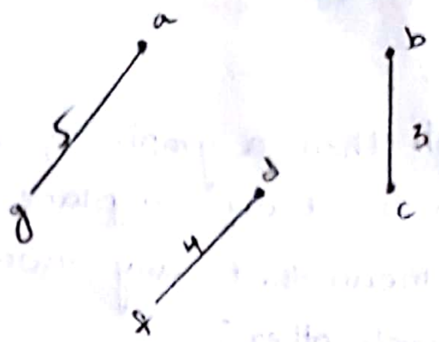
①



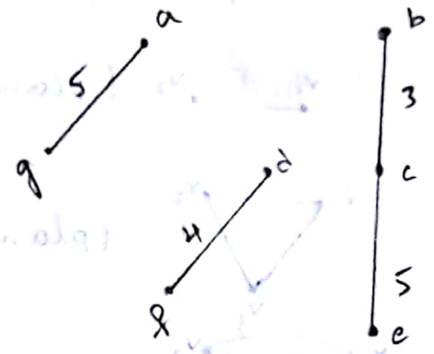
②



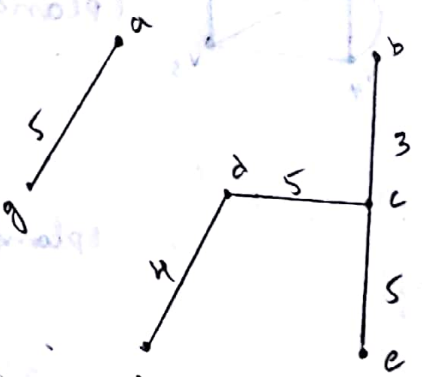
(3)



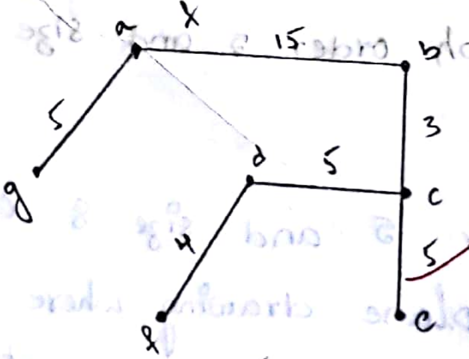
(4)



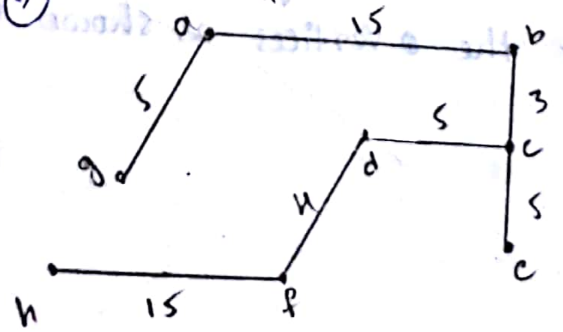
(5)



(6)



(7)



Total cost is = 15+15+5+5+5+4+3

= 52

Planar Graph :-

Let 'G' be a graph then a graph G is said to be planar graph it can be drawn on plane with out any cross overs : Crossovers mean that any two edges should not intersect to each other.

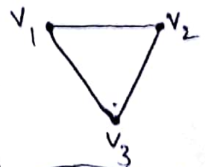
Eg:-

1. K_1 

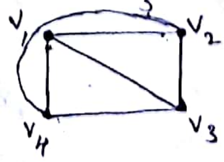
(planar graph)

2. K_2 

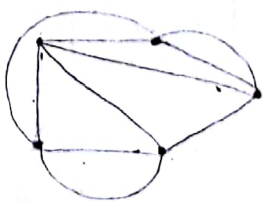
(planar graph)

3. K_3 

(planar graph)

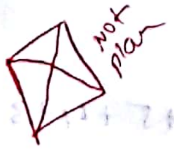
4. K_4 

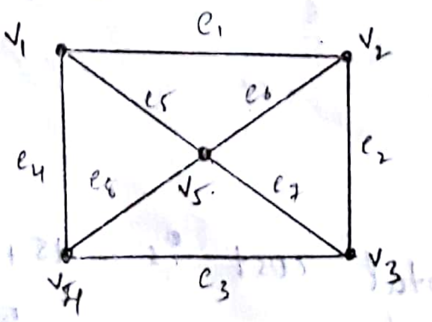
(planar graph)

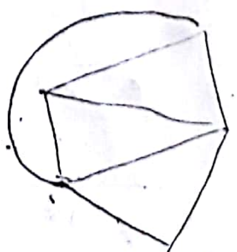
5. $K_{2,3}$ 
(planar graph)

*** ① V.IMP Show that a graph order 5 and size 8 is a planar.

Sol A graph order 5 and size 8 can be represented by a plane drawing where the edges meet only at the vertices as shown in

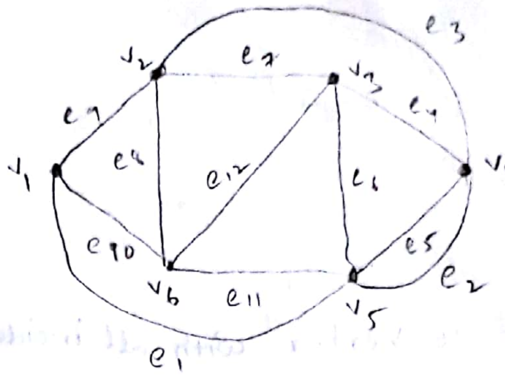






$\therefore$ Order = 5
Size = 8

2) Show that graph of order 6 and size 12 is a planar.



A graph order 6 and size 12 can be represented by a plane drawing where the edges meet only at the vertices as shown in the above figure

order = 6

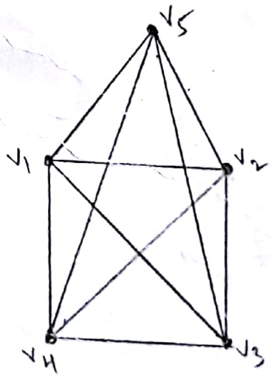
size = 12

Imp. 14M

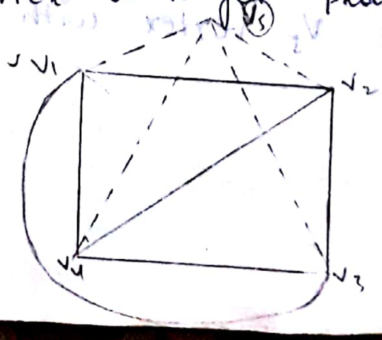
3. which of the following graphs have the property that removal of any vertex and all edges incident with that vertex produce a planar graph.

i) K_5 2) K_6 3) $K_{3,3}$ 4) $K_{3,4}$

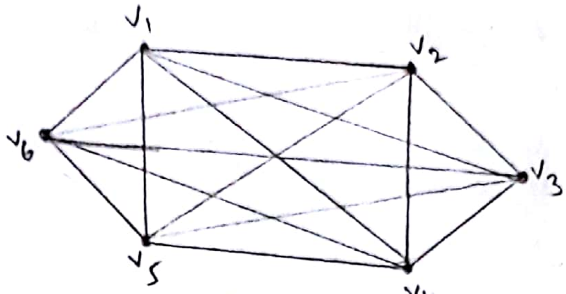
Sol i) K_5



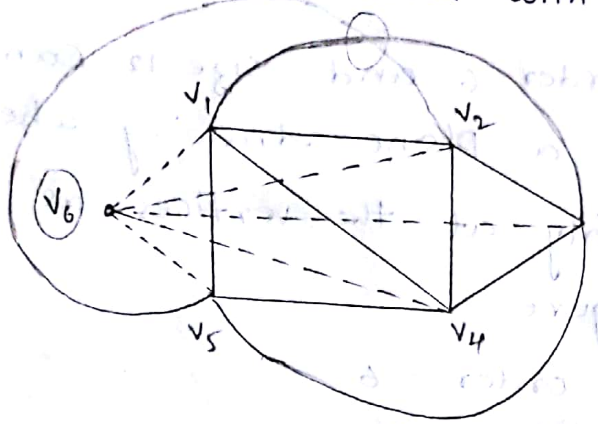
Let us remove vertex v_5 and all edges incident with that vertex to get planner graph as shown below



2) K_6

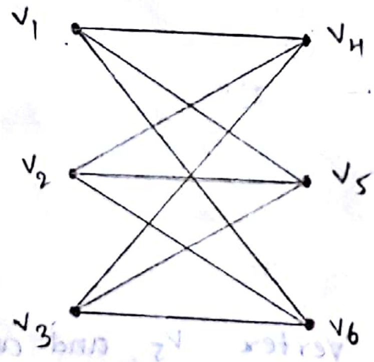


Let us remove v_6 vertex with all incident edges

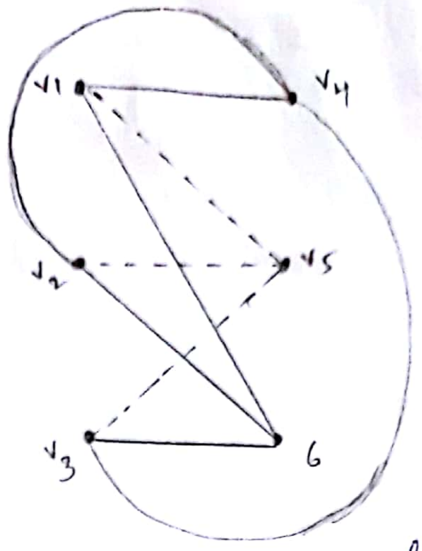


The above graph resulting after the removal of any vertex K_6 and all edges incident with the vertex cannot be a plannar graph.

3) $K_{3,3}$

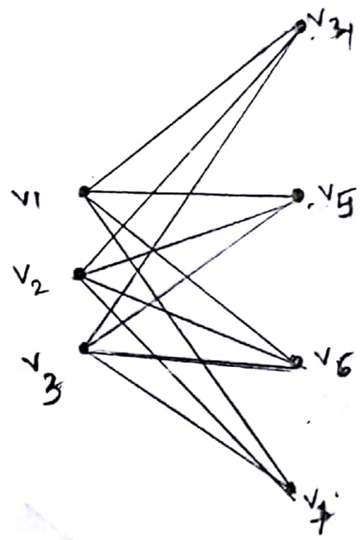


Let us Remove v_5 vertex with all incident edges

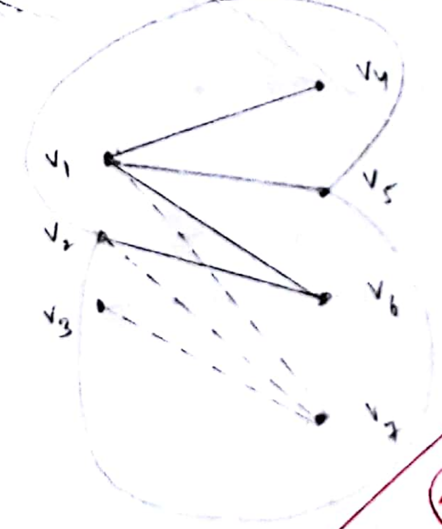


So it is a planar graph

4) $K_{3,4}$



Let us remove v_7 vertex with all incident edges



Graph Theory And Applications

1. Basic concepts, Isomorphism and sub graph
4. Multi graph
5. Euler circuits
6. Hamiltonian graphs
7. Chromatic Numbers

① Basic Concepts :-

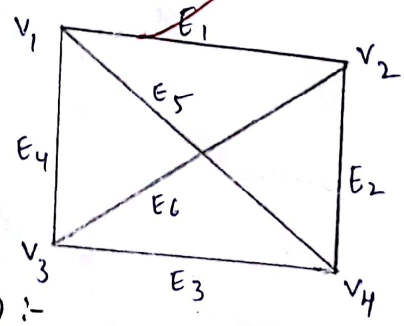
Graph :- A graph is a set of vertices and edges and it can be represented by $G = (V, E)$

Where V = Vertices

E = Edges

Connected graph :- A graph G is said to be connected graph if there is atleast one path in between every pair of vertices is called "connected graph" other wise it is called as "disconnected graph."

Connected graph :-

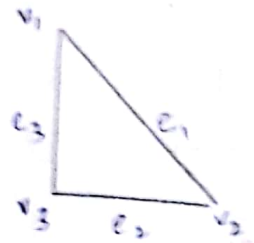


3+3+3+3
12

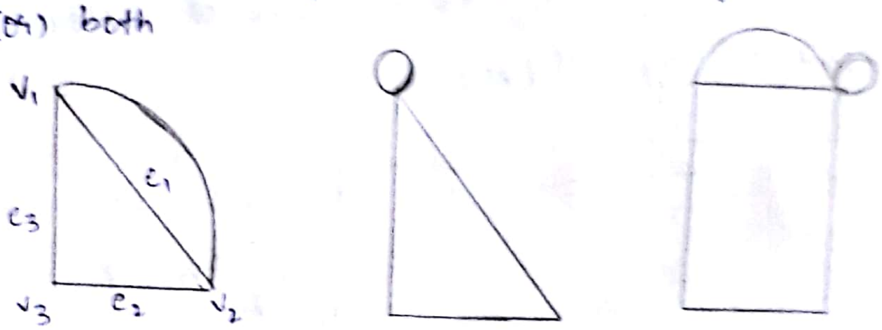
dis connected graph :-

Vertices

Simple graph :- A graph 'G' is said to be a simple graph if it does not containing either parallel edges (or) self loops.



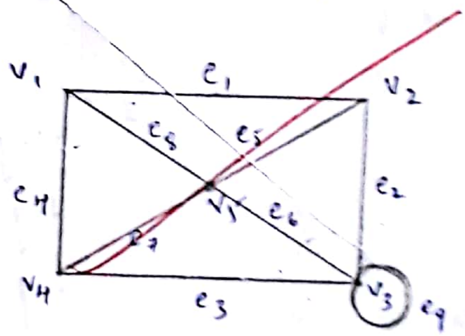
General graph :- A graph is said to be a general graph if it containing either parallel edges (or) self loops (or) both



parallel | self loop | Both

Degree of vertex :- The number of edges incident on vertex with the self loop counted twice is called as degree of vertex it can represented by $D(v_i)$ (or) $d(v_i)$.

Eg:-



$d(v_1) = 3$

$d(v_2) = 3$

$d(v_3) = 5$.

$d(v_4) = 3$

$d(v_5) = 4$

$d(v_i) = d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5)$

$= 3 + 3 + 5 + 3 + 4$

$= 18$

Scanned by CamScanner

$$= \sum_{i=1}^{|V|} d(v_i) = 18$$

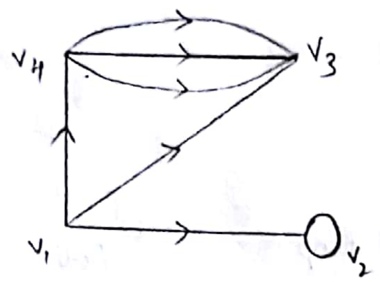
E = no. of edges

$$\boxed{\sum_{i=1}^{|V|} d(v_i) = 2|E|}$$

In degree & out degree :-

In degree of vertex :- The number of edges incident to vertex is called as indegree of vertex. it is represented by $d^+(v_i)$

Eg:-



in deg $(v_1) = 0$

in deg $(v_2) = 2$

in deg $(v_3) = 4$

in deg $(v_4) = 1$

Out degree of vertex :- The number of edges incident vertex is called as out degree of vertex it is represented by $d^-(v_i)$

out deg $(v_1) = 3$

out deg $(v_2) = 1$

out deg $(v_3) = 0$

out deg $(v_4) = 3$

total deg $(v_1) = 3$

deg $(v_2) = 3$

deg $(v_3) = 4$

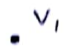
deg $(v_4) = 4$

Null graph :- A graph G is a said to be null graph if $|E| = 0$, $|V| \neq 0$

Eg:-

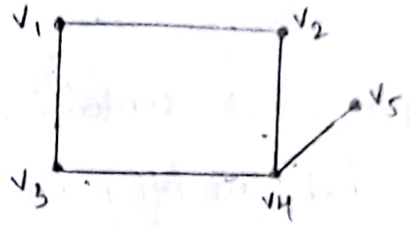
Isolated vertex :- A vertex is said to be isolated vertex the degree is '0'

Eg:-



$deg(v_1) = 0$

Eg:-

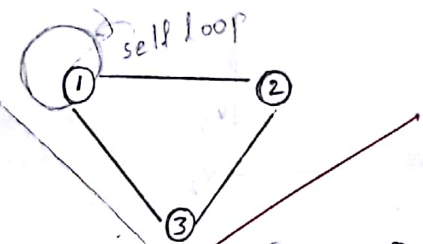


$deg(v_5) = 0$

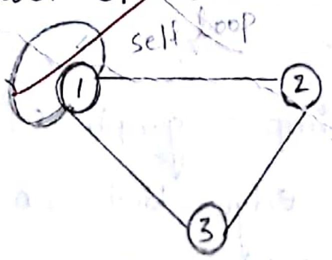
pendent vertex :- A vertex is said to be pendent vertex if it having the degree '1'.

$deg(v_1) = 1$
$deg(v_2) = 1$

Self loop :- An edge which is between a vertex and itself is called as a self loop

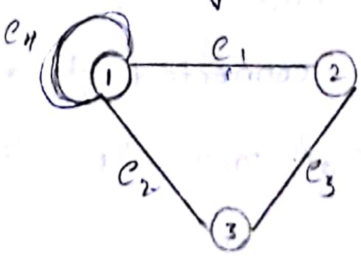


Order :- Number of vertices in a graph is called a order



Order = 3

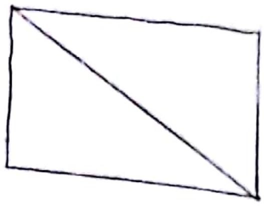
Size :- Number of edges in a graph is called a size.



Size = 4

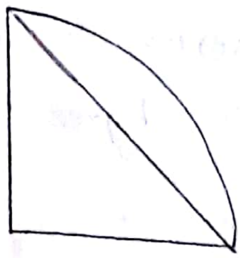
Loop-free :- A graph which does not containing loops is called a loop-free

Eg:-



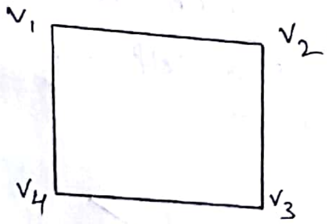
Multi graph :- A graph which containing multiple edges but no loops is called multi graph.

Eg:-

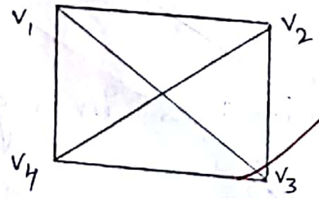


Regular graph :- A graph 'G' is said to be a regular graph it contains all vertices have same degree

Eg:-1



Eg:-2



Complete graph :- A simple graph of order it ≥ 2 in which there is an edge between every pair of vertices is called complete graph (or) full graph.

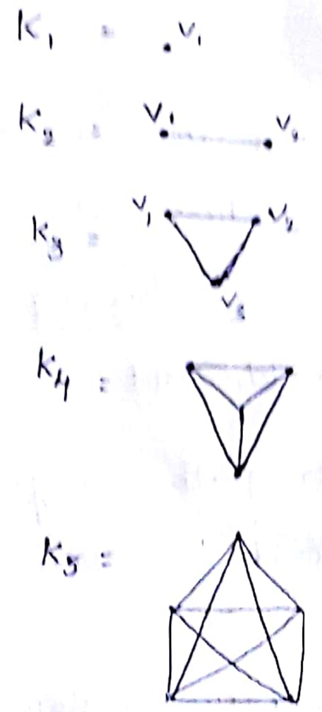
(Or)

* A graph 'G' is said to be complete graph, which every vertex is connected to all vertices

* Complete graph of order 'n' it can be represen by K_n

Where n is the number of vertices

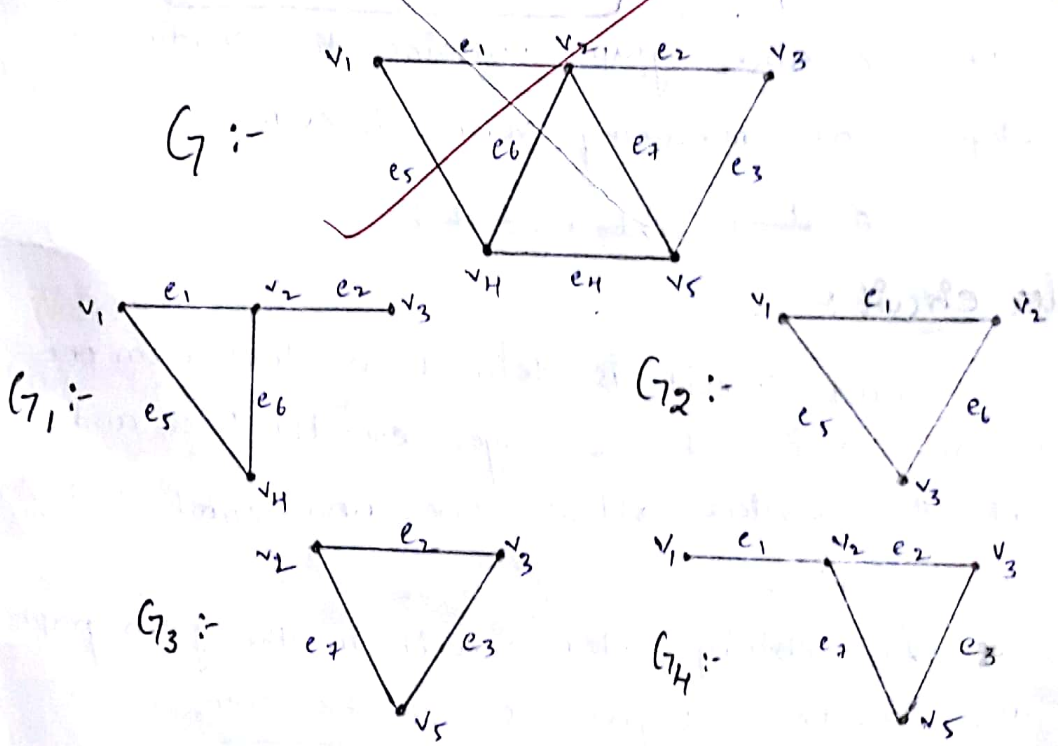
Eg:



K_n every vertex degree should be $(n-1)$

K_n having $\frac{n(n-1)}{2}$ edges

5) **Sub graph** :- Let $G(V,E)$ and $G(V_1,E_1)$ any two graphs then a graph is said to be sub graph of G. if $V_1 \leq V$, $E_1 < E$ and an edge is associated with pair of vertices and associated with same edge in G



Eulerian graph :-

Eulerian Graph :-

1. Euler path :-

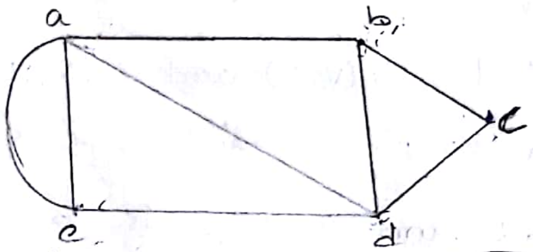
* Euler path is defined as visit all the edges exactly once and visit all the vertices atleast once.

* In the euler path the starting vertex and ending vertex ~~in the~~ should not be same.

* To obtain the euler path for given graph, all the vertices degree should be even, except two vertices whose degree is odd.

* To obtain the euler path we can start from one odd degree vertex and end with another odd degree vertex.

Eg:-



Euler path : $\boxed{b - \cancel{c} - d - b - a - d - e - a -}$ e

In the above graph contains two vertices is odd degree and remaining vertices is even.

~~e-d-a-e-d-b-c-d-b~~

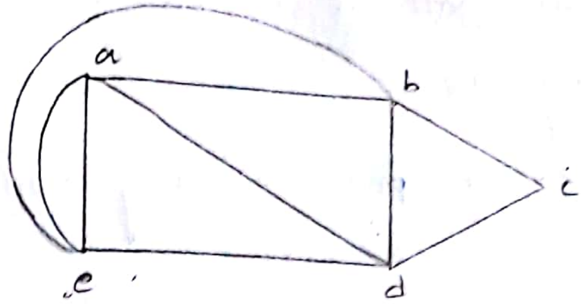
2. Euler circuit :-

Euler circuit is defined as start from one vertex and visit all the edges exactly once and visit all the vertices atleast once and comback to starting vertex.

* To obtain euler circuit in the given graph all the vertices degree should be even.

* In euler circuit starting vertex and ending vertex should be same

Eg :-



$\sum_{i=1}^{|V|} d(v_i) = 2|E|$

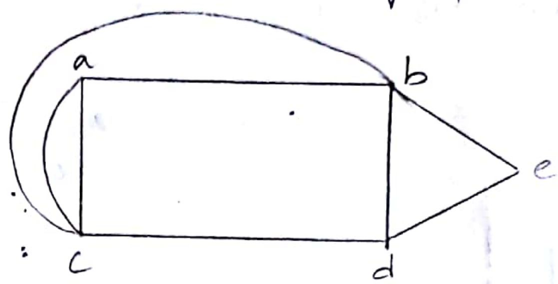
Euler circuit :-

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

Eulerian (or) Euler graph :-

If any graph consist euler circuit then this graph is called as Eulerian graph.

Eg :-



* Find the graph consists No. of edges if it having 16 vertices of degree 2.

Sol :- For non directed graph, we have $\sum_{i=1}^{|V|} d(v_i) = 2|E|$

Sum of degree of all vertices $= 2|E|$

$2 \times 16 = 2|E|$

$|E| = 16.$

Note :-1 An Euler circuit contain even no. of vertices and odd no. of edges.

Note :- 2 An Euler circuit contain odd number of vertices and even no. of edges.

Hamiltanian graph

Eg:- Hamiltanian Path :-

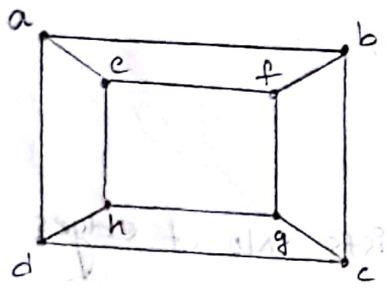
1. Hamiltanean path is defined as visit all the vertices exactly once.

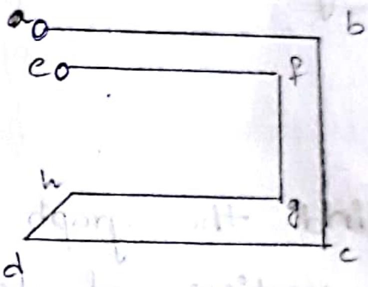
2. In Hamiltanean path starting vertex and ending vertex should not be same.

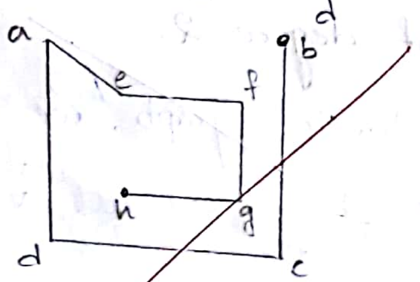
3. If the given graph having 'n' vertices then hamiltanean path having (n-1) edges here 'n' is number of vertices in a given graph.

4. The hamiltaneon path terminal vertices degree should be one and remaining all the vertices degree should be two.

Eg :-







path :- b - c - d - a - e - f - g - h

Exp :- In the above graph containing 8 vertices and 7 edges

Hamiltaneon circle :-

* Hamiltaneon cycle is defined as start from one vertex and visit all the vertices exactly once and come back to starting vertex.

* In hamiltaneon cycle starting vertex and ending vertex should be same

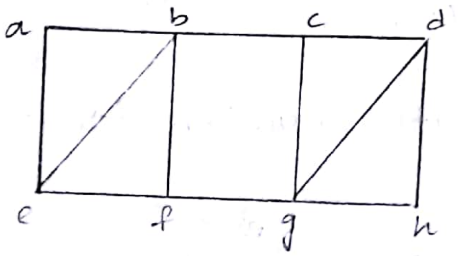
* In hamiltaneon cycle all the vertices degree should be two.

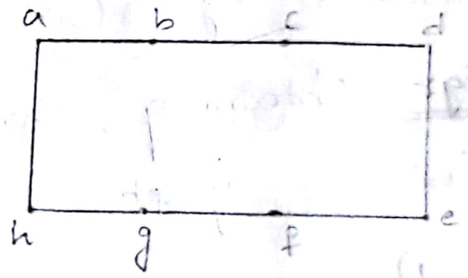
* If the given graph having 'n' vertices and 'm' edges then hamiltaneon cycle of given graph 'n' vertices and 'm' edges remaining (m-n) edges can be scipped from the given graph.

* In given graph the vertex degree is 'n' then the hamiltaneon cycle of degree should be two i.e., remaining (n-2) edges should be scipped from the given graph.

$12 - 8 = 4$

Eg:-





H.C = a-b-c-d-h-g-f-e-a

Note :- For any graph if there is a haimaltaneon cycle then they should be a haimaltaneon path.

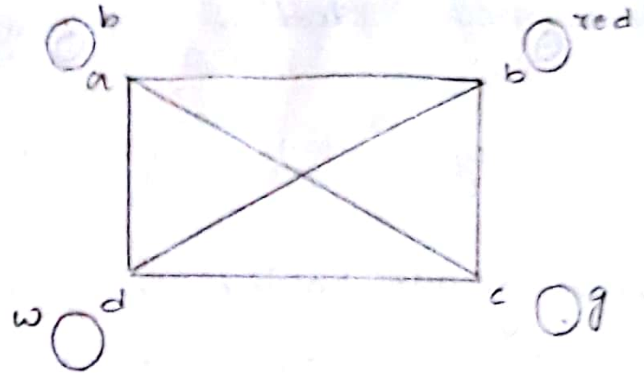
* A graph is said to be hamiltaneon graph it contains hamiltaneon cycle.

Chromatic Number :-

colouring :- Assigning (or) painting colours to all the vertices in a given graph.

Proper colouring :- Assiagning (or) painting colours to all the vertices such that no two adjusent vertices have same colour

Eg:-



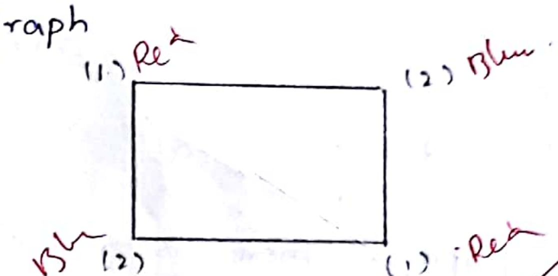
Chromatic number :-

The minimum number of colours required for proper colouring is called as chromatic number.

Here proper colouring means assigning (or) painting colours to all the vertices such that no to adjacent vertices have same colour it is denoted by $\chi(G)$ $\chi(G)$

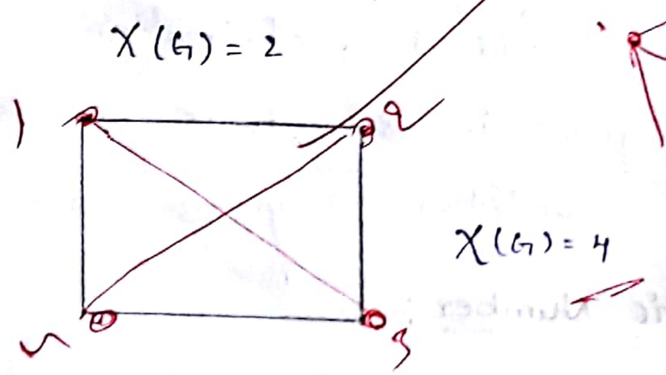
Eg:- Obtaining chromatic numbers for the following graph

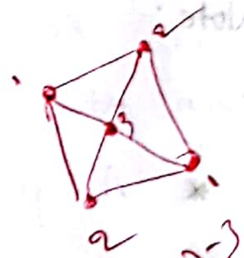
1)



$\chi(G) = 2$

2)

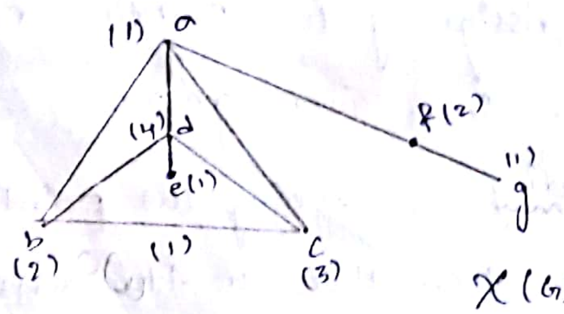




$\chi(G) = 4$

$\chi(G) = 3$

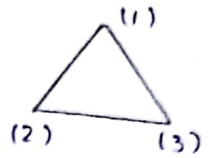
3)



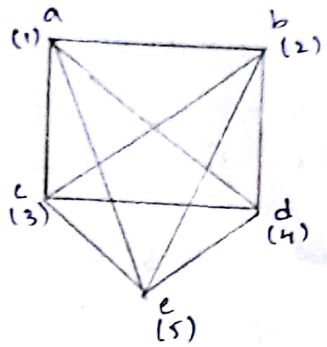
$\chi(G) = 4$

4) 

$\chi(G) = 1$

5) 

$\chi(G) = 3$

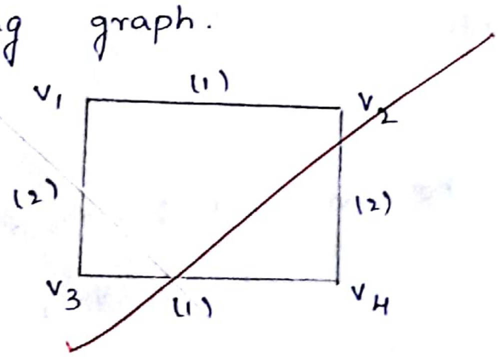
6) 

$\chi(G) = 5$

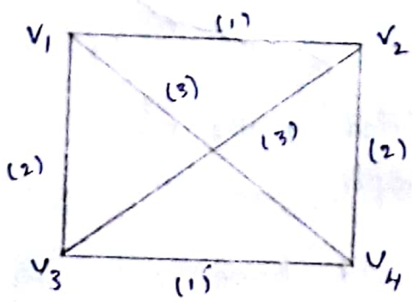
Chromatic number of edges :-

The minimum number of colours required for assigning colours to all the edges such that No two adjesent edges have the same colour.

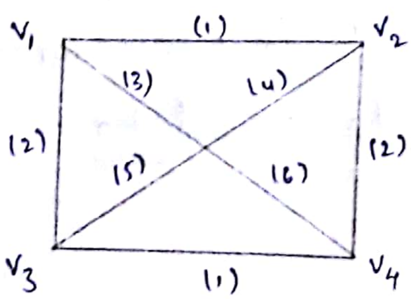
Find the chromatic number of the edges to the following graph.

1) 

no. of chromatic $\chi(G) = 2$

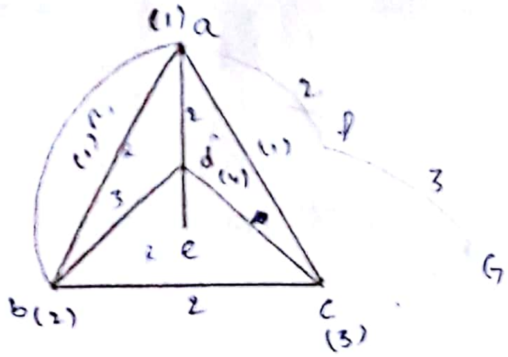
2) 

$\chi(G) = 3$

3) 

$\chi(G) = 6$

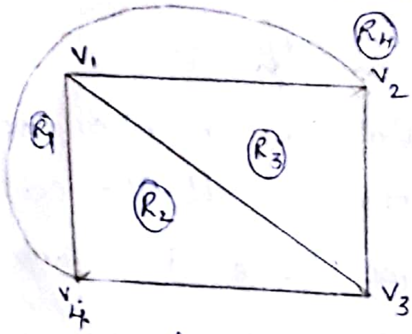
4)



$\chi(G) =$

Region :-

Every planar graph can divides the plane into regions. region can be characterised by set of edges forming its boundaries is called as "region".



In the above planar graph, The planar graph can divide the plane into four regions that are R_1, R_2, R_3, R_4.

The region R_1 boundaries are : V_1-V_2, V_2-V_4, V_4-V_1.

The region R_2 " " : V_1-V_4, V_4-V_3, V_3-V_1

The region R_3 " " : V_1-V_2, V_2-V_3, V_2-V_3

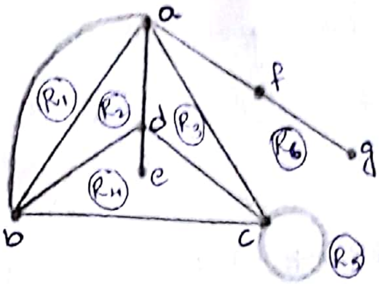
The region R_4 " " : V_2-V_4, V_4-V_3, V_3-V_2

Exterior region :- The position of the plane which is line outside of the planar graph is called as exterior region

Every planar graph should have only one exterior region

In above planar graph R_4 is a exterior region the remaining region R_1, R_2, R_3 are interior regions.

Obtain boundaries of region in following graph :-



Sol

R_1 : a - b - a

R_2 : a - b - d - a

R_3 : a - d - c - a

R_4 : b - a - e - d - c - b

R_5 : c - c

R_6 : g - f - a - b - c - c - a - f - g

Note :-

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

If 'G' is a planar graph, n is a number of vertices, r is a region then the no. of regions is

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

$$= 11 - 7 + 2$$

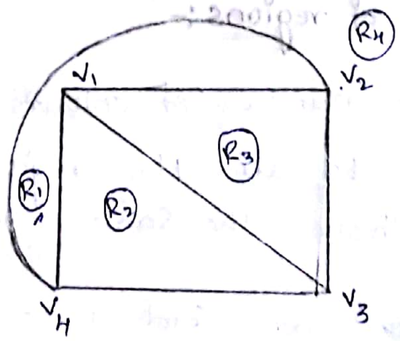
$$= 4 + 2$$

$$R = 6$$

II **Degree of region :-**

The length of the boundaries of a region (no. of edges in boundaries of region) is called as degree of region

Eg :-



$R_1 = 3$

Scanned by CamScanner

Sol

Degree of region $(R_1) = 3$

Degree of region $(R_2) = 3$

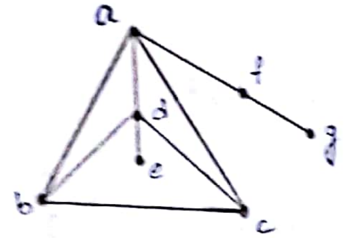
Degree of region $(R_3) = 3$

Degree of region $(R_4) = 3$

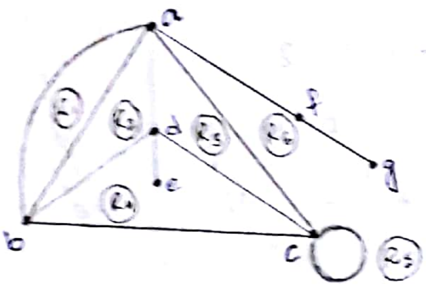
$\mathrm{Deg}(R) = \deg(R_1) + \deg(R_2) + \deg(R_3) + \deg(R_4)$

---

* obtain the degree of all regions for given planar graph.



Sol



R_1 : a - b - a $\mathrm{Deg}(R_1) = 2$

R_2 : a - d - b - a $\mathrm{Deg}(R_2) = 3$

R_3 : a - d - c - a $\mathrm{Deg}(R_3) = 3$

R_4 : b - d - e - d - c - b $\mathrm{Deg}(R_4) = 5$

R_5 : c - c $\mathrm{Deg}(R_5) = 1$

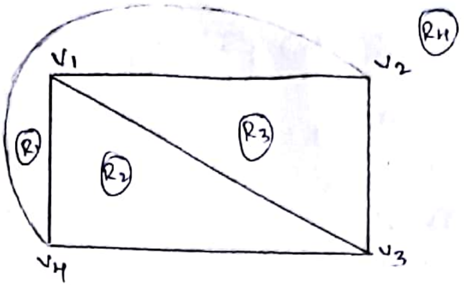
R_6 : g - f - a - b - c - c - a - f - g $\mathrm{Deg}(R_6) = 8$

Chromatic number of regions :-

Minimum number of colours required for assigning colours to all the region such that no two adjacent region have the same colour

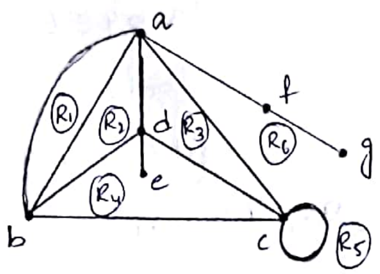
→ Two regions are said to be adjacent if these two regions can be seperated with common edge

1)



$\chi(R) = 4$

2)



$\chi(G) = 3$

Imp 14M

*** ISOMORPHISM :-

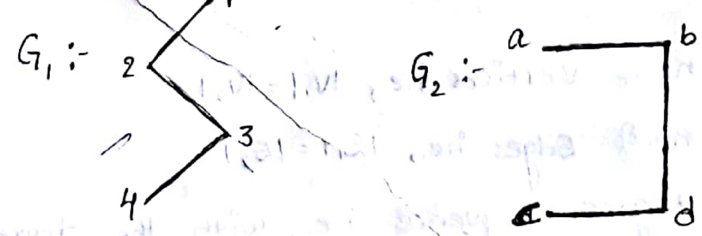
→ Consider two graphs $G_1 = (V_1, E_1)$ & $G_2 = (V_2, E_2)$ Suppose there exist a function $f : V_1 \to V_2$ such that.

i) f is one-to-one i.e., f is bijective

ii) $\{a, b\}$ is an edge in E_1, if and only if $\{f(a), f(b)\}$ is an edge in E_2 for any two elements $a, b \in V_1$

→ This should preserve the adjacency relation and consequently the corresponding vertices in G_1 and G_2 will have the same degree

Eg:- The function f is an isomorphism b/w G_1 and G_2.



Sol:-

Here $V(G_1) = \{1, 2, 3, 4\}$

$V(G_2) = \{a, b, c, d\}$

$E(G_1) = \{(1,2)\ (2,3)\ (3,4)\}$

$E(G_2) = \{(a,b)\ (b,d)\ (d,c)\}$

Scanned by CamScanner

A function $f: V(G_1) \to V(G_2)$

$f(1) = a$

$f(2) = b$

$f(3) = d$

$f(4) = c$

f is clearly one-to-one and onto

$(1,2) \in E(G_1)$ and $\{f(1), f(2)\} = \{a, b\} \in E(G_2)$

$(2,3) \in E(G_1)$ and $\{f(2), f(3)\} = \{b, d\} \in E(G_2)$

$(3,4) \in E(G_1)$ and $\{f(3), f(4)\} = \{d, c\} \in E(G_2)$

Hence G_1 and G_2 are Isomorphic

Note :-

It is practically not possible to check for all possible functions $f: V_1 \to V_2$ for example if there are 10 vertices in a graph then there are 10! such functions. It is not always an easy task to determine whether or not, to given graphs are Isomorphic however, we can prove that two graphs are not isomorphic by showing that they do not share a property that isomorphic graphs must both have, such a property is called invarient.

Defination :- A property shared by isomorphic graphs is called as isomorphism invarient we may list some of the following invarient. If G_1 and G_2 are isomorphism graphs then G_1 & G_2 have the

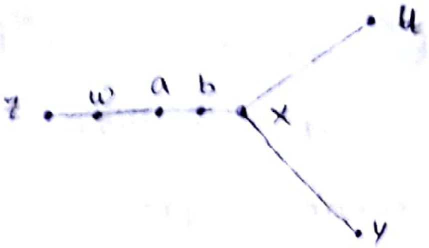
i) Same no. of vertices i.e, $|V_1| = |V_2|$

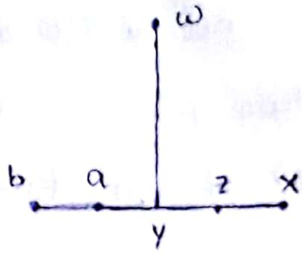
ii) Same no. of Edges i.e., $|E_1| = |E_2|$

iii) Same degree sequence i.e., with the degree of vertex v_i in G_1 is m, then the degree of vertex v_i in G_2 is m,

* If any of these quantities differ in two graphs they cannot be Isomorphic.

* However this conditions are by no means sufficient for example.





$V(G_1) = 7$

$E(G_1) = 6$

deg $(x(G_1)) = 3$

deg $(u(G_1)) = 1$

deg $(y(G_1)) = 1$

deg $(z(G_1)) = 1$

deg $(w(G_1)) = 2$

deg $(a(G_1)) = 2$

deg $(b(G_1)) = 2$

$V(G_1) = 6$

$E(G_1) = 5$

deg $(a(G_2)) = 2$

deg $(b(G_2)) = 1$

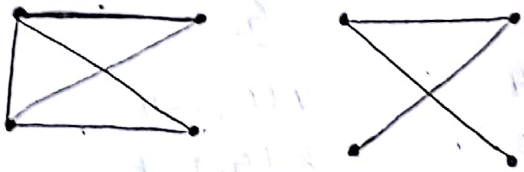
deg $(x(G_2)) = 1$

deg $(y(G_2)) = 3$

deg $(z(G_2)) = 2$

deg $(w(G_2)) = 1$

1) Show that two graphs are isomorphic or not.



Sol:-

$\underline{G_1}$

$V(G_1) = 4$

$E(G_1) = 5$

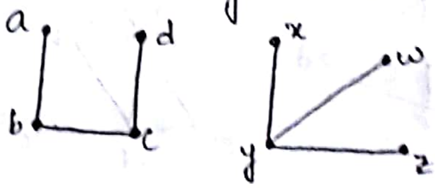
$\underline{G_2}$

$V(G_2) = 4$

$E(G_2) = 3$

So the above graphs are not isomorphic because in first graph have 5 edges and second graph have 3 edges

2) Show that the two graphs are Isomorphic or not.



Sol

$\underline{G_1}$

$V(G_1) = 4$

$E(G_1) = 3$

deg $(a) = 1$

deg $(b) = 2$

deg $(c) = 2$

deg $(d) = 1$

$\underline{G_2}$

$V(G_2) = 4$

$E(G_2) = 3$

deg $(x) = 1$

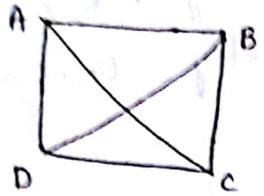
deg $(y) = 3$

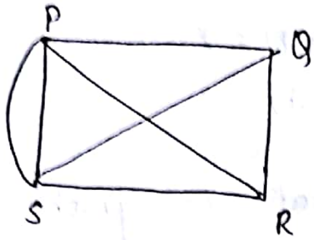
deg $(z) = 1$

deg $(w) = 1$

* It is not a isomorphic because observe that there are two pendent vertex in G_1 and three pendent vertex in G_2 (pendent means degree 1).
* As such under any one-to-one correspondence b/w the vertices and edges of the two graphs, the adjacency of vertices is not preserved.
* The degree sequence is also not sufficient.

3) Show that the following graphs are not isomorphic





Sol

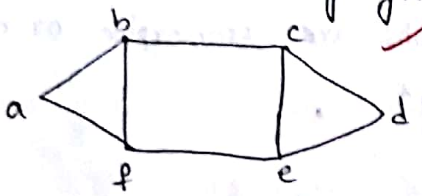
G_1 G_2

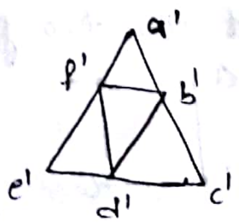
$V(G_1) = 4$ $V(G_2) = 4$

$E(G_1) = 6$ $E(G_2) = 7$

We observe that a graph G_1 has 4 vertices and 6 edges and a graph G_2 has 4 vertices and 7 edges i.e., $E(G_1) \neq E(G_2)$. As such one to one correspondence b/w the edges is not possible. Hence the two graphs are not isomorphic.

4) Show that the following graphs are isomorphic or not





Sol

G_1 G_2

$V(G_1) = 6$ $V(G_2) = 6$

$E(G_1) = 9$ $E(G_2) = 9$

deg (a) = 2 deg (a') = 2

deg (b) = 3 deg (b') = 4

deg (c) = 4 deg (c') = 2

deg (d) = 2 deg (d') = 4

deg (e) = 3 deg (e') = 2

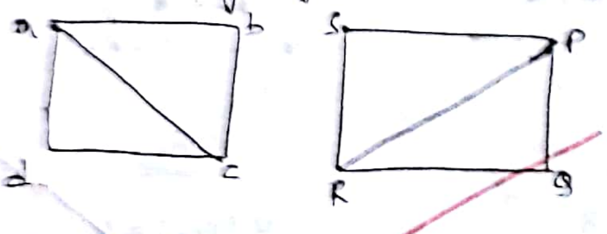
deg (f) = 4 deg (f') = 4

deg (G_1) = {2, 2, 3, 3, 4, 4}

deg (G_2) = {2, 2, 2, 4, 4, 4}

We observe that the first graph G_1 has two vertices of deg 4 i.e deg (c), deg (f) and second graph G_2 has three vertices of deg 4 i.e., deg (b'), deg (d'), deg (f'). Therefore there cannot be any one-to-one correspondence b/w the vertices & edges of the two graphs. As such the given graphs are not isomorphic.

5) Show that following graphs are isomorphic or not



Sol

G_1 G_2

$V(G_1) = 4$ $V(G_2) = 4$

$E(G_1) = 5$ $E(G_2) = 5$

deg (a) = 3 deg (P) = 3

deg (b) = 2 deg (Q) = 2

deg (c) = 3 deg (R) = 3

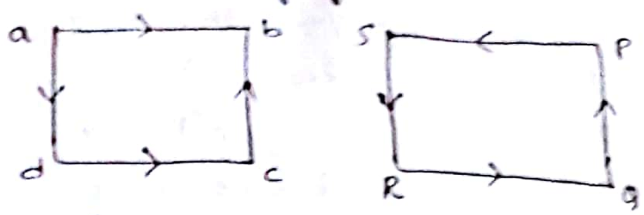
deg (d) = 2 deg (S) = 2

deg (G_1) = {3, 3, 2, 2}

deg (G_2) = {3, 3, 2, 2}

So it is isomorphic

(6) Show that the following graphs are isomorphic or not



Sol:-

<u>G_1</u>

$V(G_1) = 4$

$E(G_1) = 4$

deg (a) = 2

deg (b) = 0

deg (c) = 1

deg (d) = 1

<u>G_2</u>

$V(G_2) = 4$

$E(G_2) = 4$

deg (P) = 1

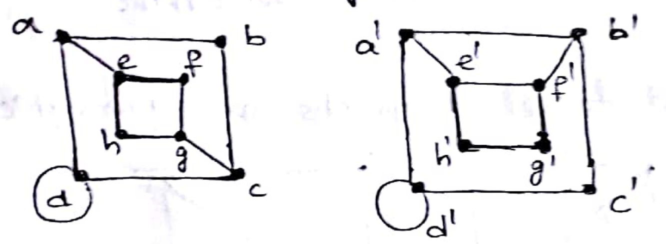
deg (Q) = 1

deg (R) = 1

deg (S) = 1

degree Sequence is not sufficient so it is not isomorphic

(7) Show that the following graphs are isomorphic or not



Sol

<u>G_1</u>

$V(G_1) = 8$

$E(G_1) = 11$

deg (a) = 3

deg (b) = 2

deg (c) = 3

deg (d) = 4

deg (e) = 3

deg (f) = 2

deg (g) = 3

deg (h) = 2

<u>G_2</u>

$V(G_2) = 8$

$E(G_2) = 11$

deg (a') = 3

deg (b') = 3

deg (c') = 2

deg (d') = 4

deg (e') = 3

deg (f') = 3

deg (g') = 2

deg (h') = 2

$\deg(G_1) = \{2, 2, 2, 3, 3, 3, 3, 4\}$

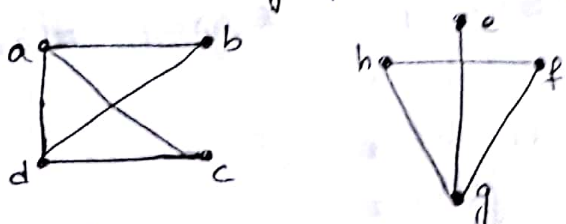
$\deg(G_2) = \{2, 2, 2, 3, 3, 3, 3, 4\}$

In graph G_1 four vertices of degree is 3 i.e., deg(a), deg(c), deg(e), deg(g) but these are not adjacent.

In graph G_2 four vertices of degree 3 i.e., deg(a'), deg(b'), deg(e'), deg(f') but these are adjacent.

∴ Therefore the two graphs are not isomorphic.

(8) Show that the two graphs are isomorphic (or) not.



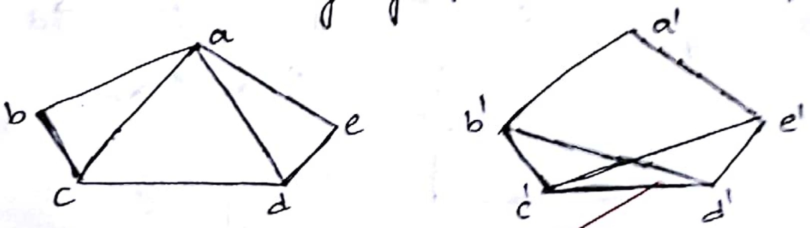
G_1 G_2

$V(G_1) = 4$ $V(G_2) = 4$

$E(G_1) = 5$ $E(G_2) = 4$

∴ The two graph are not isomorphic

(9) Find the following graphs are isomorphic (or) not



Sol

G_1 G_2

$V(G_1) = 5$ $V(G_2) = 5$

$E(G_1) = 7$ $E(G_2) = 7$

deg(a) = 4 deg(a') = 2

deg(b) = 2 deg(b') = 3

deg(c) = 3 deg(c') = 3

deg(d) = 3 deg(d') = 3

deg(e) = 2 deg(e') = 3

~~deg (A~~

$deg(G_1) = \{4, 2, 2, 3, 3\}$

$deg(G_2) = \{2, 3, 3, 3, 3\}$

It is not isomorphic because the degree sequence is not sufficient.

10) Find whether the following graphs are Isomorphic or not

Sol

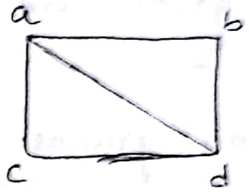
G_1	G_2	G_1	G_2
$V(G_1) = 4$	$V(G_2) = 4$	$V(G_1) = 4$	$V(G_2) = 4$
$E(G_1) = 4$	$E(G_2) = 4$	$E(G_1) = 4$	$E(G_2) = 5$

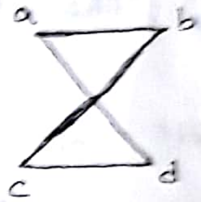
deg(a) = 2	deg(a') = 2
deg(b) = 1	deg(b') = 2
deg(c) = 3	deg(c') = 2
deg(d) = 2	deg(d') = 2

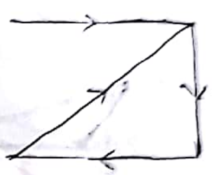
It is not isomorphic

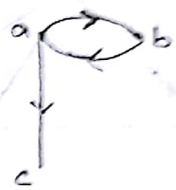
11) Find the adjacency matrix of each of the following.

a) 

b) 

c) 

d) 

e) 

a)

	1	2	3	4
1	0	1	1	0
2	1	0	0	1
3	1	0	0	0
4	0	1	0	0

b)

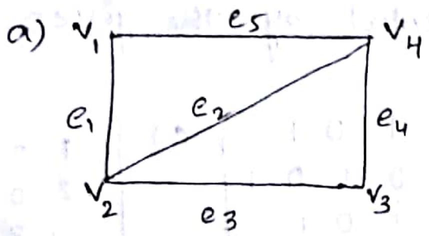
	a	b	c	d
a	0	1	1	1
b	1	0	0	1
c	1	0	0	1
d	1	1	1	0

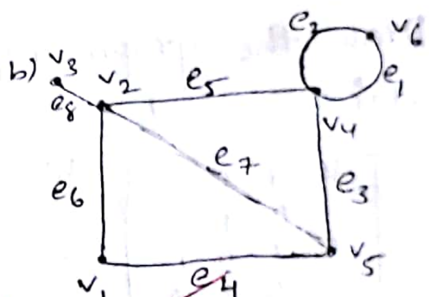
c)
$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ c & 0 & 1 & 0 & 1 \\ d & 1 & 0 & 1 & 0 \end{array}$$

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

e)
$$\begin{array}{c|ccc} & a & b & c \\ \hline a & 0 & 1 & 1 \\ b & 1 & 0 & 0 \\ c & 0 & 0 & 0 \end{array}$$

12) find the incidence matrix of the following graphs

a) 

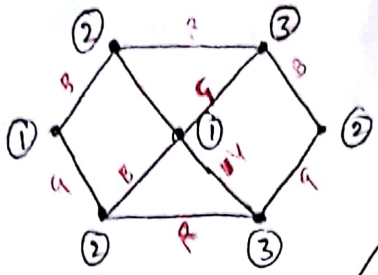
b) 

Sol

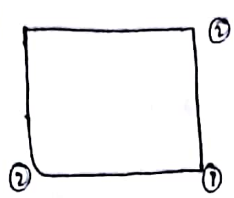
a)
$$\begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{array}\begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

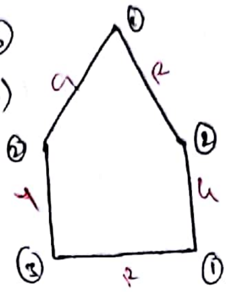
b)
$$\begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array}\begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

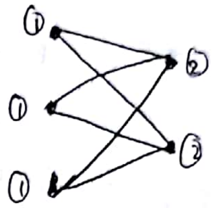
(13) Find the chromatic number of each graph

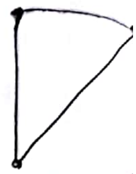
a)


b)


c)


d)


e)


f)


Sol

a) chromatic number $\chi(G) = 3$

b) Chromatic number $\chi(G) = 4$

c) chromatic number $\chi(G) = 2$

d) chromatic number $\chi(G) = 3$

e) chromatic number $\chi(G) = 2$

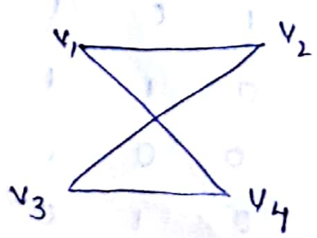
f) Chromatic number $\chi(G) = 3$

14) Draw the graph represented by the given adjacency matrix

a) $\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$ b) $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$ c) $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

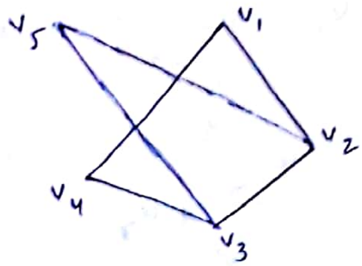
Sol

a) $\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$



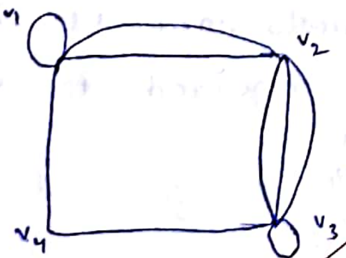
b)

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



c)

$$\begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\ \hline \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$



15) Draw the directed graph represented by the given adjacency matrix

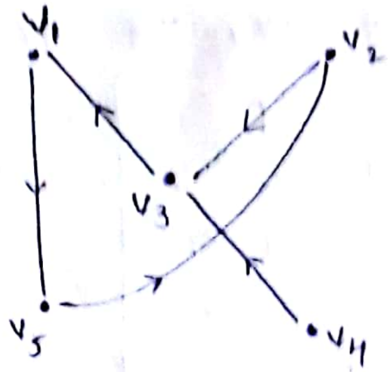
a) $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$

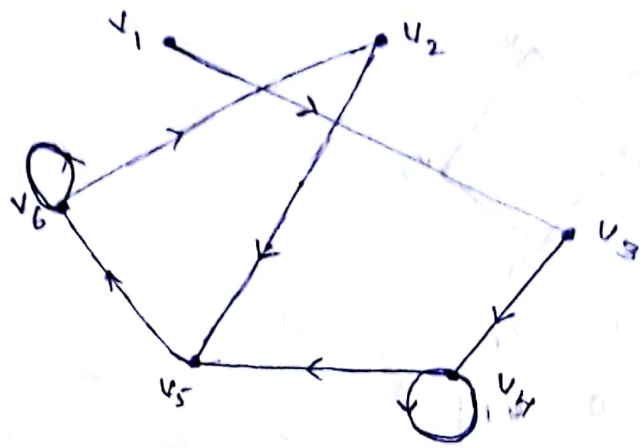
Sol

a)

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

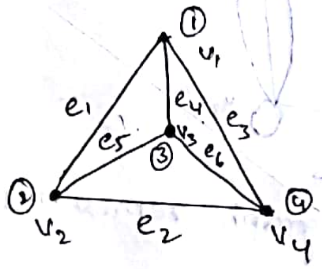


b)



16) Find the no. of vertices and no. of edges and no. of regions of each map in figure given below and verify Euler's formula. And also find minimum no. of colours to paint required to each map

a)



Sol

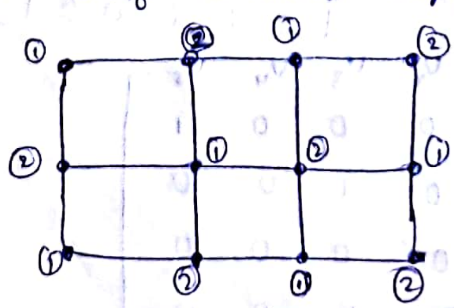
No. of vertices = 4

No. of edges = 6

No. of regions = 4

Minimum no. of colours required = 4

b)



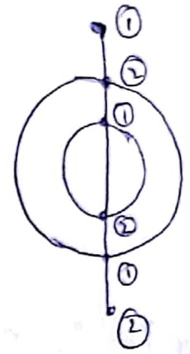
No. of vertices $v = 12$

No. of edges $= e = 17$

No. of regions $= r = 7$

minimum no. of colours required $\chi = 3$

c)



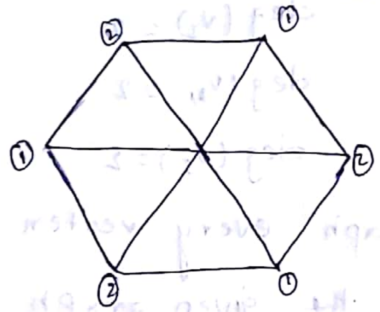
No. of vertices $: v = 6$

No. of edges $= e = 9$

No. of regions $= r = 5$

minimum no. of colours required $\chi = 2$

d)



No. of vertices $= v = 6$

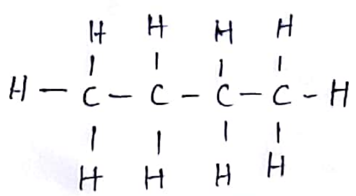
No. of edges $= e = 9$

No. of regions $= r = 7$

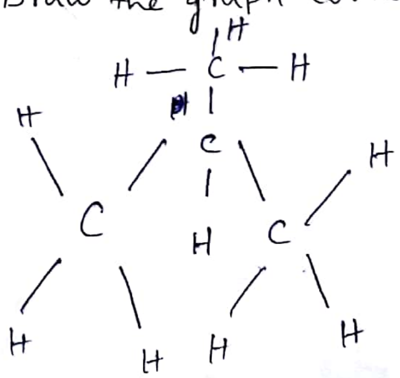
Minimum no. of colours required $\chi = 2$

(17) There are two different chemical molecules with formula C_4H_{10} (Isobutane). Draw the graph corresponding to this molecule.

C_4H_{10} graph:-

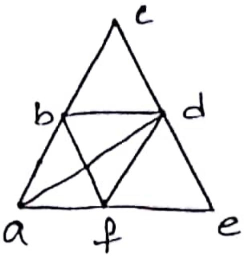


(or)



Determine which of the following graphs contain an Eulerian circuit if it does, then find the Eulerian ~~circuits~~ graph

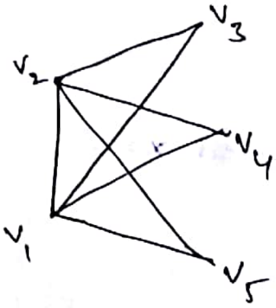
a)



Sol

a is not an even so it is not a Eulerian graph.

b)



$deg(v_1) = 4$

$deg(v_2) = 4$

$deg(v_3) = 2$

$deg(v_4) = 2$

$deg(v_5) = 2$

Sol

In the above graph every verten contain an even degree so the given graph is Eulerian graph.

$v_5 - v_1 - v_4 - v_2 - v_3 - v_1 - v_5$ this is the path for Eulerian circuit