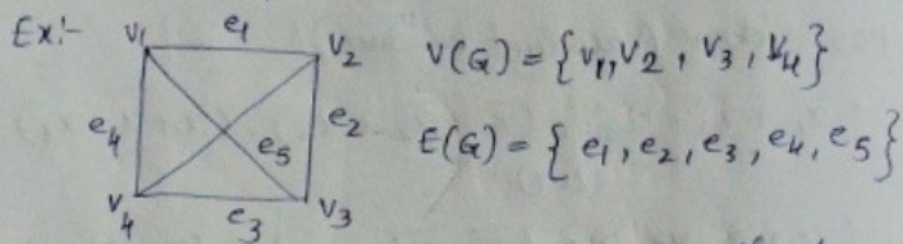


21/12/23

Unit - 5 Graph theory

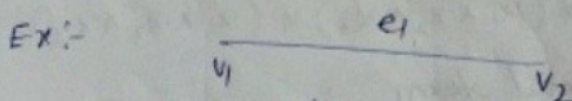
Basic concepts, isomorphism and sub graph, trees and their properties, spanning trees, directed trees, binary trees, planar graph, Euler's formula, multigraph & Euler's circuit, Hamiltonian graph, chromatic number, the four-color problem.

Graph :- A graph $G(V, E)$ consist of a non empty set of vertices (V) and edges (E) may or may not be empty of a graph. We denote a graph as $G(V, E)$ or simple G .



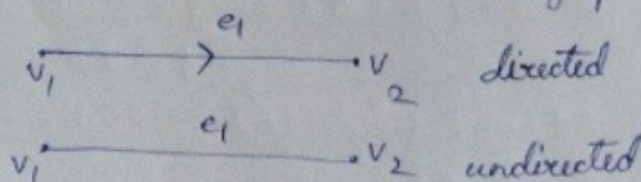
Note:- we can represent number of different diagrams (graphs) for same number of vertices.

- Any pair of vertices which are connected by an edge is called adjacent vertices.



where v_1 and v_2 are adjacent vertices

- In a graph $G(V, E)$ and edge is associated with an ordered pair of vertices is called directed edge and while and edge is associated with unordered pair of vertices is called undirected graph

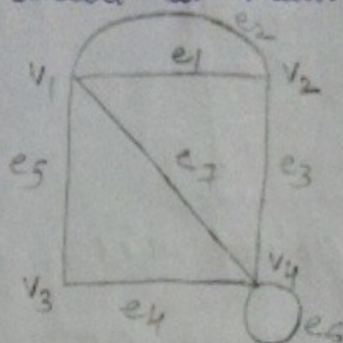


Types of graphs:-

1. multi graph:

Any graph which contains parallel and loop edges is called as multi graph

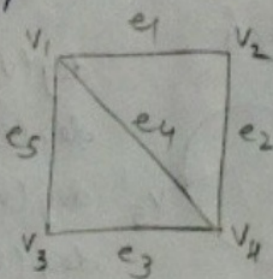
EX:-



2. simple graph:

A graph with no parallel and loop edges is called as simple graph

EX:-



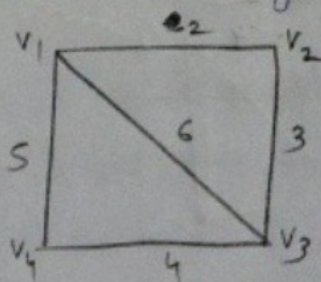
3. Trivial graph:-

A graph with single vertex having no edges is called trivial graph.

EX:- $\bullet v_1$

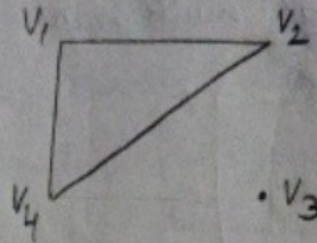
4. weighted graph:- A graph which contains edges with weight is called as weighted graph.

EX:-



5. Isolated vertex:

In a graph all vertices are connected except one vertex, that vertex is called as isolated vertex.

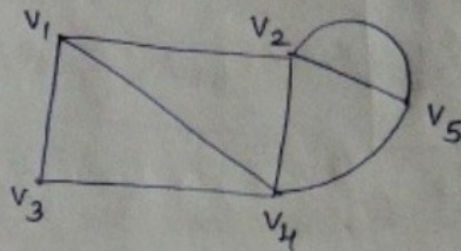


Here v_3 is isolated vertex

6. Degree of graph:-

Let v be a vertex in a graph G , then the degree of vertex v in G is number of edges incident with vertex v , it can be denoted as $\deg(v)$

Ex:- undirected graph



$$\deg(v_1) = 3$$

$$\deg(v_2) = 4$$

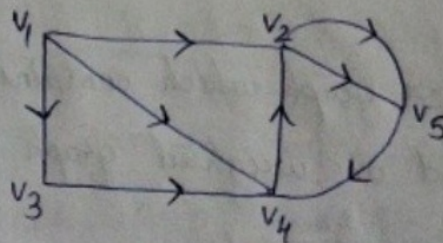
$$\deg(v_3) = 2$$

$$\deg(v_4) = 4$$

$$\deg(v_5) = 3$$

Indegree / outdegree

directed graph



In order

$$\deg(v_1) = 0$$

$$v_2 = 2$$

$$v_3 = 1$$

$$v_4 = 3$$

$$v_5 = 2$$

out order

$$\deg(v_1) = 3$$

$$v_2 = 2$$

$$v_3 = 1$$

$$v_4 = 1$$

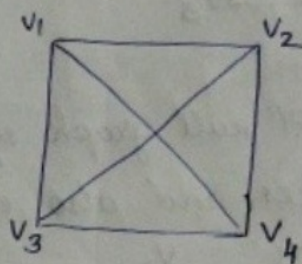
$$v_5 = 1$$

7. Complete graph:-

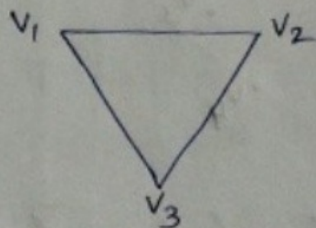
A simple graph in which every pair of distinct vertices are adjacent is called as complete graph (K_n).

EX:- (K_4)

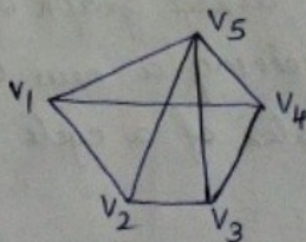
$n = \text{vertices}$



(K_3)

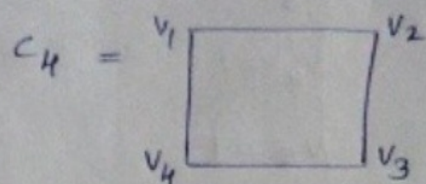
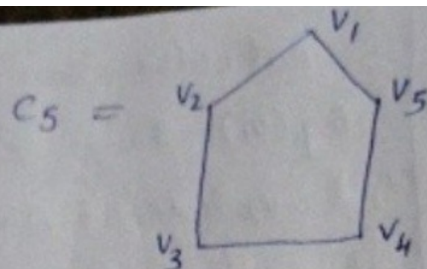


(K_5)

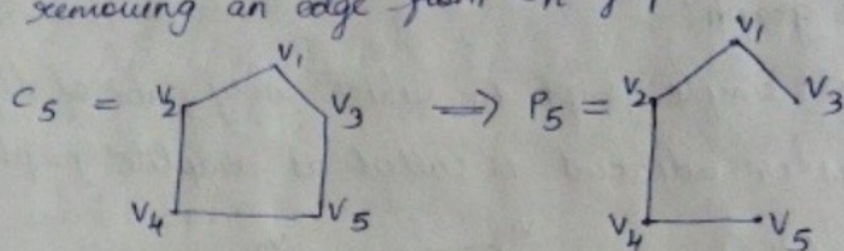


8. Cycle graph:- A cycle graph of order n is connected graph whose edges form a cycle of length n and denoted by (C_n).

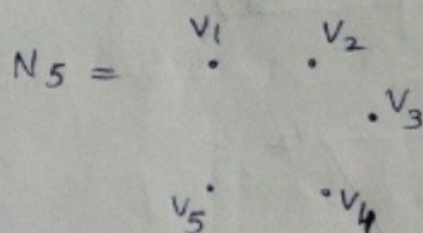
Ex:-



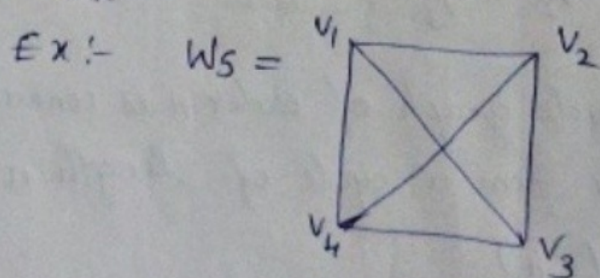
9. Path graph: A path graph of order n is obtained by removing an edge from C_n graph denoted by P_n .



10. Null graph:- A null graph of order n is a graph with n vertices and also no edges denoted by N_n .



11. wheel graph:- A wheel graph of order n is obtained by joining a new vertex called hub to each vertex of a cycle graph denoted by W_n .



EX:- 1. Draw a diagram of a graph $G=(V, E)$ in each of the following cases.

1. $V = \{A, B, C, D\}$

$E = \{(A, B), (A, C), (A, D), (C, D)\}$

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

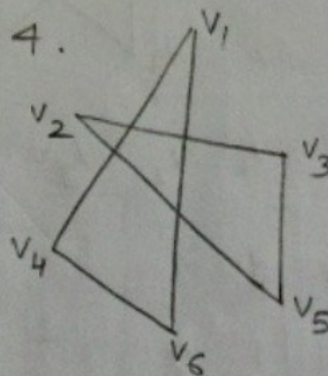
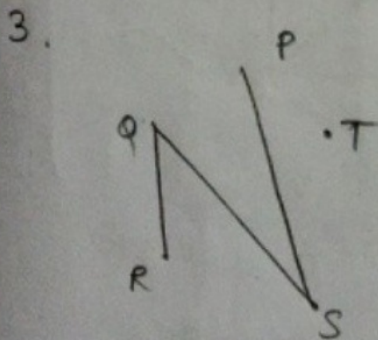
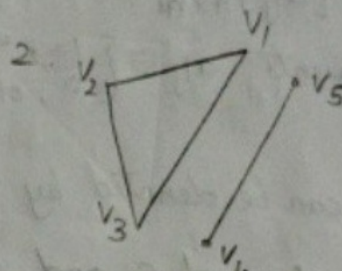
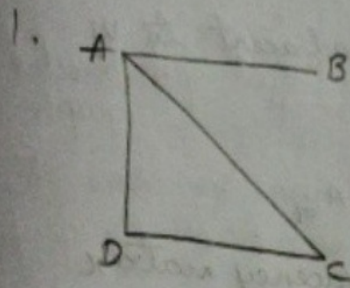
$E = \{(V_1, V_2), (V_1, V_3), (V_2, V_3), (V_4, V_5)\}$

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

$E = \{(P, S), (Q, R), (Q, S)\}$

4. $V = \{V_1, V_2, V_3, V_4, V_5, V_6\}$

$E = \{(V_1, V_4), (V_1, V_6), (V_4, V_6), (V_3, V_2), (V_3, V_5), (V_2, V_5)\}$



28/12/23

→ Representation of graphs:
Matrix representation of graph:
In this representation we have 3 types:

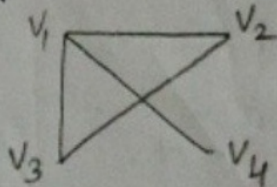
1. Adjacency
2. Incidence matrix
3. Path matrix

1. Adjacency:

Let $G(V, E)$ be a simple graph with n vertices ordered from $v_1 \dots v_n$ then the adjacency matrix $A = [a_{ij}]_{n \times n}$ of G is an $n \times n$ matrix defined by elements $a_{ij} = \begin{cases} 1, & \text{when } v_i \text{ is adjacent to } v_j \\ 0, & \text{otherwise} \end{cases}$ [undirected graph]

It can be denoted by $A(G)$ or A_G .

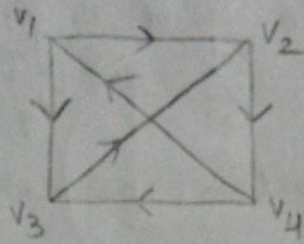
Ex:- A graph G and find adjacency matrix



	v_1	v_2	v_3	v_4
v_1	0	1	1	1
v_2	1	0	1	0
v_3	1	1	0	0
v_4	1	0	0	0

In case of directed graph $a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \text{ edges} \\ 0, & \text{otherwise} \end{cases}$

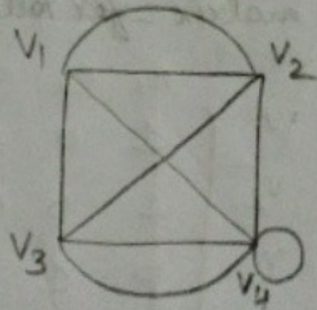
ex:- Find adjacency matrix for following diagram



$$A_G = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \end{matrix}$$

$$\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

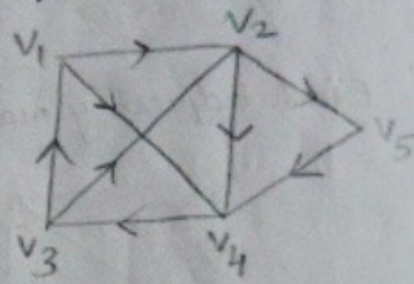
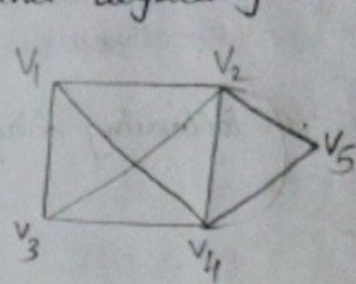
In case of multigraph



$$G = \begin{matrix} & v_1 & v_2 & v_3 & v_4 \end{matrix}$$

$$\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 2 & 1 \end{bmatrix}$$

Find adjacency matrix of following graph.



2. Incidence matrix

Let G be a graph with n vertices and m edges

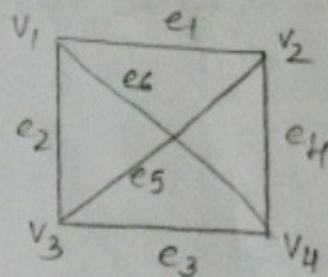
$$V = \{v_1, v_2, \dots, v_n\} \text{ and } E = \{e_1, e_2, \dots, e_m\}$$

define $n \times m$ matrix

$$\text{Here } I_G = [m_{ij}]_{n \times m}$$

$$\text{where } m_{ij} = \begin{cases} 1, & \text{where } v_i \text{ is incident with } e_j \\ 0, & \text{otherwise} \end{cases}$$

Ex:- Find Incidence matrix for following graph.

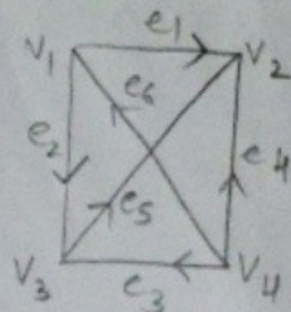


$$\begin{matrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

In case of directed graph

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

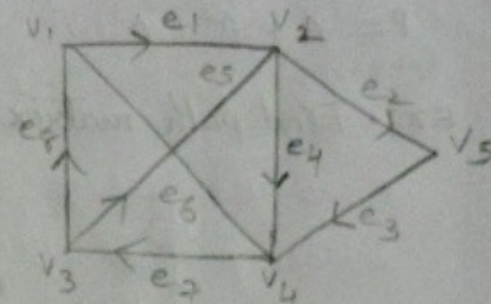
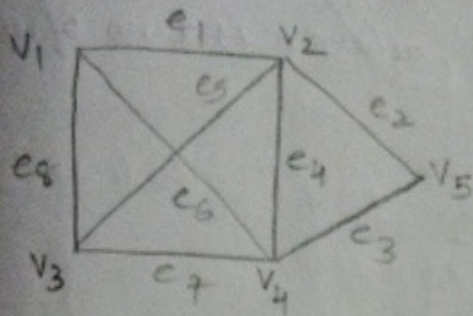
Ex:- Find Incidence matrix



outgoing = 1 (initial)
incoming = -1 (final)

$$\begin{array}{c}
 v_1 \\
 v_2 \\
 v_3 \\
 v_4
 \end{array}
 \begin{bmatrix}
 e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
 1 & 1 & 0 & 0 & 0 & -1 \\
 -1 & 0 & 0 & -1 & -1 & 0 \\
 0 & -1 & -1 & 0 & 1 & 0 \\
 0 & 0 & 1 & 1 & 0 & 1
 \end{bmatrix}$$

Find ~~ad~~ Incidence matrix for the following

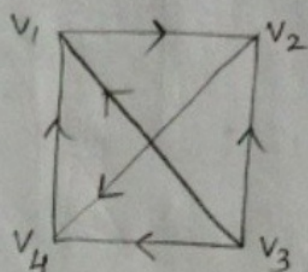


3. path matrix:

let G be a simple digraph having no parallel directed edges & $V = \{v_1, v_2, v_3, \dots, v_n\}$ be its vertex set and $n \times n$ matrix $P = [P_{ij}]_{n \times n}$ is given by $P_{ij} = \begin{cases} 1, & \text{if there is a path from } v_i \text{ to } v_j \\ 0, & \text{otherwise} \end{cases}$

$P = A + A^2 + A^3 + \dots + A^n$, where n is a no. of vertices

Ex:- Find path matrix



$n = 4$ vertices

$$P = A + A^2 + A^3 + A^4$$

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

$$A^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (0+0+0+0) & (0+0+0+0) & (0+0+0+0) & (0+1+0+0) \\ (0+0+0+1) & (0+0+0+0) & (0+0+0+0) & (0+0+0+0) \\ (0+0+0+1) & (1+0+0+0) & (0+0+0+0) & (0+1+0+0) \\ (0+0+0+0) & (1+0+0+0) & (0+0+0+0) & (0+0+0+0) \end{bmatrix}$$

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

$$A^3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (0+0+0+1) & (0+0+0+0) & (0+0+0+0) & (0+0+0+0) \\ (0) & (1+0+0+0) & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

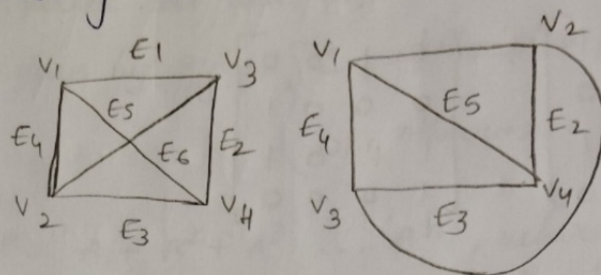
$$A^4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & 0 & 1 \\ 1 & 1 & 0 & 2 \\ 4 & 4 & 0 & 4 \\ 2 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

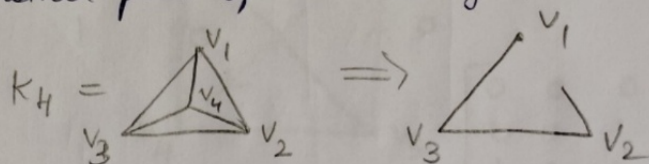
Sat

planar Graph:-
A graph G is said to be planar if it can be drawn in the plane without its edges crossing otherwise G is non planar.

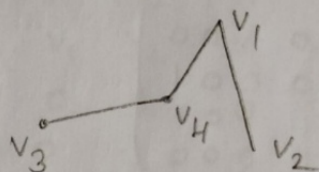


Sub graph:-

A sub graph is part of a graph which is another part of another graph.

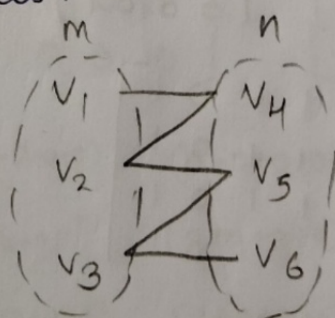


Complement:- $\bar{G} = K - G$



Bipartial graphs:-

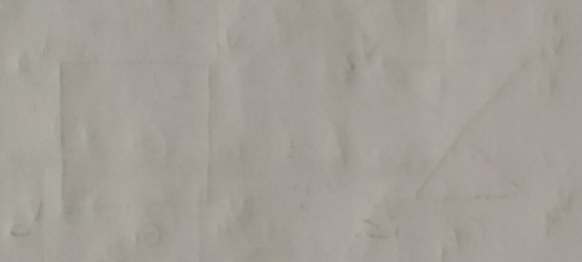
It is an undirected graph whose set of vertices of can be divided into two sets and each vertex is atleast connected with two vertices.



complete bipartial graph:-

It is a bipartial graph in which every vertex

Ex: draw the following graph and isomorphic or not



Sol: 1. Both graphs must have same no of vertices

$$G_1 = 4 \text{ vertices}, G_2 = 4 \text{ vertices}$$

It is satisfied

2. Both graphs must have same no of edges

$$G_1 = 4, G_2 = 4 \text{ edges}$$

It is satisfied

3. Both graphs must have same degree

degree

$$G_1 = (2, 2, 2, 2), G_2 = (2, 2, 2, 2)$$

It is satisfied

there it is isomorphic graph

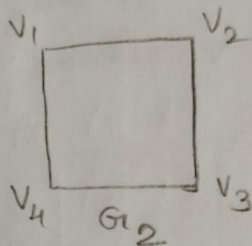
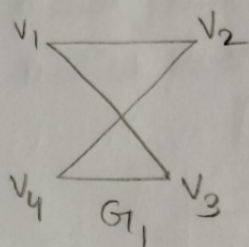
2/1/24

Isomorphism/Isomorphic graph:-

Two graphs G_1 & G_2 are Isomorphic if and only if they have.

1. the same no of vertices
2. same no of edges
3. same degree sequence.

Ex:- show the following graph are isomorphic or not.



sol:- 1. Both graphs must have same no of vertices

$$G_1 = 4 \text{ vertices} \quad G_2 = 4 \text{ vertices.}$$

It is satisfied

2. Bothe graphs must have same no of edges

$$G_1 = 4, \quad G_2 = 4 \text{ edges}$$

It is satisfied

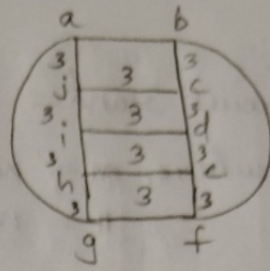
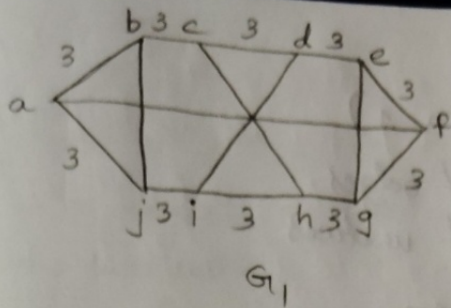
3. Bothe graphs must have same degree sequence

$$G_1 = (2, 2, 2, 2) \quad , \quad G_2 = (2, 2, 2, 2)$$

v_1 edges passing

It is satisfied

hence it is Isomorphic graph



sol: 1. Both must have same no of vertices = a, b, c, ...

$$G_1 = 10, G_2 = 10$$

It is satisfied.

2. Both must have same no of edges

$$G_1 = 15, G_2 = 15$$

satisfied

3. must have same degree sequence

$$G_1 = (3, 3, 3, 3, 3, 3, 3, 3, 3, 3)$$

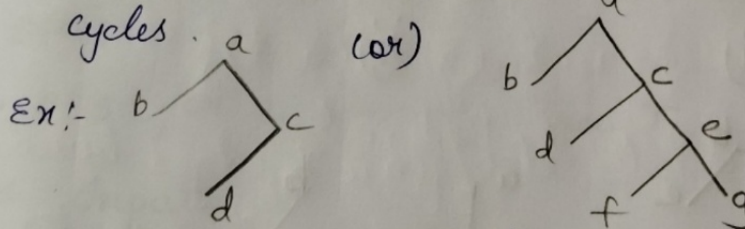
$$G_2 = (3, 3, 3, 3, 3, 3, 3, 3, 3, 3)$$

satisfied

It is a Isomorphic graph.

⇒ spanning tree :-

A tree is a graph G which is connected or no cycles.



let G be a connected graph. A subgraph T of G is it contains all the vertices of G called as a spanning tree of G . It satisfies the following conditions.

1. T is tree
2. T contains all the vertices of graph

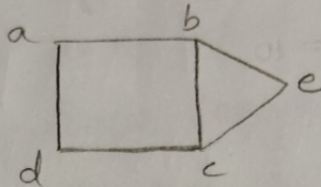
note:-

• circuit rank = $m - (n - 1)$

where m = no. of edges

n = no. of vertices

Ex:- Find spanning tree



sol we need to find out circuit rank of given graph

m = no of edges = 6

n = vertices = 5

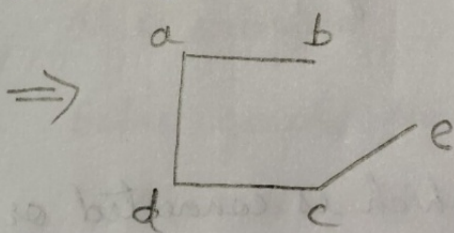
circuit rank = $m - (n - 1)$

$= 6 - (5 - 1)$

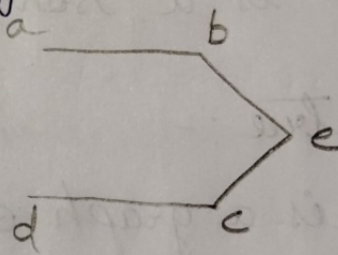
$= 6 - 4$

$= 2$

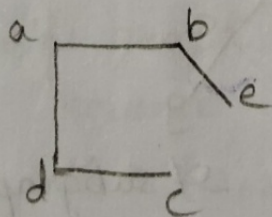
we can obtain the spanning tree from above graph by removing 2 edges ~~from~~



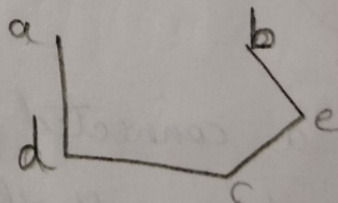
i)

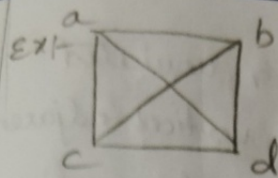


ii)



iii)





$$m = \text{edges} = 6$$

$$n = \text{vertices} = 4$$

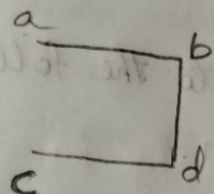
$$\text{rank} = m - (n - 1)$$

$$= 6 - (4 - 1)$$

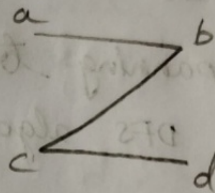
$$= 6 - 3$$

$$= 3$$

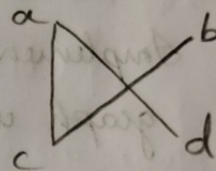
remove 3 edges



i)



ii)



iii)

⇒ DFS & BFS Algorithm:-

we can use 2 algorithms for obtaining spanning tree those are.

1. Depth first search
2. Breadth first search.

1 DFS:-

• DFS for a spanning tree T.

1. Input

A connected graph $G(V_1, V_2, \dots, V_n)$ vertices.

2. Output

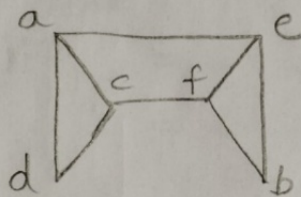
A spanning tree T

step i: [visit a vertex] let v_1 be the root of T
and set equal to set $L = v_1$

step ii: [find an unexamined edge & unvisited vertex to L] for all vertices adjacent to L, choose the edge and also confirm that adding edge to T. that does not produce a cycle.

step iii: [backtracking] If x is the parent of L
In T set $L = x$ and apply step 2.
DFS implement has stack.

Ex:- Implement spanning tree to the following graph using DFS algorithm.



Sol:- All vertices in above graph in an order

a, b, c, d, e, f

2. Now we select vertex a has root node of T

• a

3. find the adjacent edges of vertex 'a'

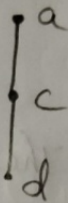
these are $\{(a, c), (a, e), (a, d)\}$

4. Now we select set of $\{(a, c)\}$ edges based on ~~the~~ order c is next working vertex



5. find the adjacent edge of vertex 'c'
those are
 $\{(c, a), (c, d), (c, f)\}$

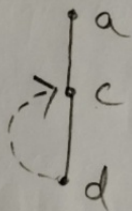
6. Now we select set of $\{c, d\}$ edge based on order d that does not produce cycle



7. Now working vertex node is d. find the adjacent edge of vertex 'd',
those are

$$\{(d, a), (d, c)\}$$

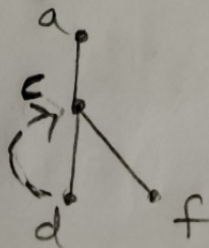
8. No more edges is selected now move to parent node of d.



9. c is the next working node. The adjacent vertex of 'c' those are

$$\{(c, a), (c, d), (c, f)\}$$

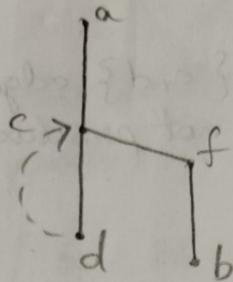
10. Now we select set of $\{c, f\}$ edge based on order f that does not produce cycle.



11. Next working vertex node ~~is~~ is f. Find the adjacent edge ~~are~~ of vertex 'f'
those are

$$\{(f, b), (f, c), (f, e)\}$$

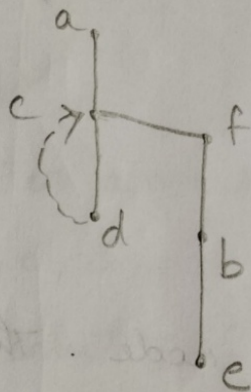
11. Now we select set of $\{f, b\}$ edge based on order b . & does not produce cycle.



12. Next working node is b . The adjacent edges of vertex ' b ' are

$$\{(b, e), (b, f)\}$$

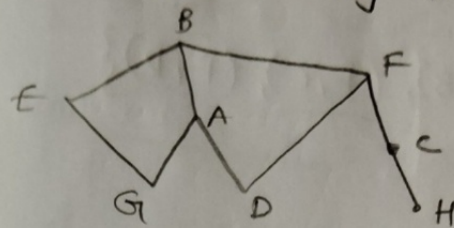
13. Now we select set of $\{b, e\}$ edge based on order e . & does not produce cycle.



14. The spanning tree of above graph by using DFS algorithm. The result order is

$$R.O = \{a, c, d, f, b, e\}$$

2. EX:- Implement Spanning Tree to the following graph using DFS algorithm.



sol:- All vertices in above graph in an order

A B C D E F G H

• A $\{(A, B), (A, D), (A, G)\}$

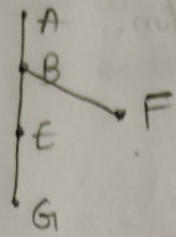
• A
|
B $\{(B, A), (B, E), (B, F)\}$

• A
|
B
|
E $\{(E, B), (E, G)\}$

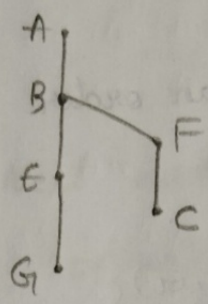
• A
|
B
|
E
|
G $\{(G, A), (G, E)\}$

• A
|
B
|
E
|
G $\{(E, B), (E, G)\}$

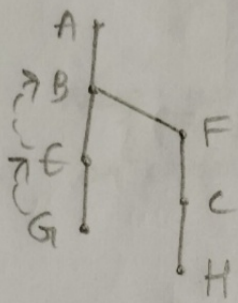
• A
|
B
|
E
|
G $\{(B, A), (B, E), (B, F)\}$



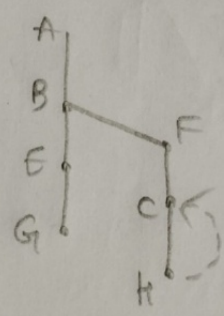
$$\{(F, B), \underline{(F, C)}, (F, D)\}$$



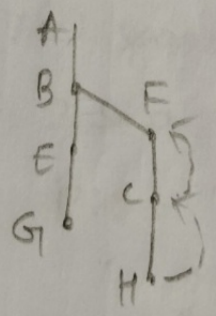
$$\{(C, F), \underline{(C, H)}\}$$



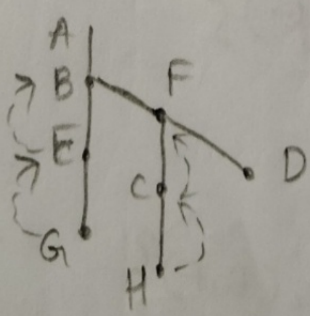
$$\{H, \underline{C}\}$$



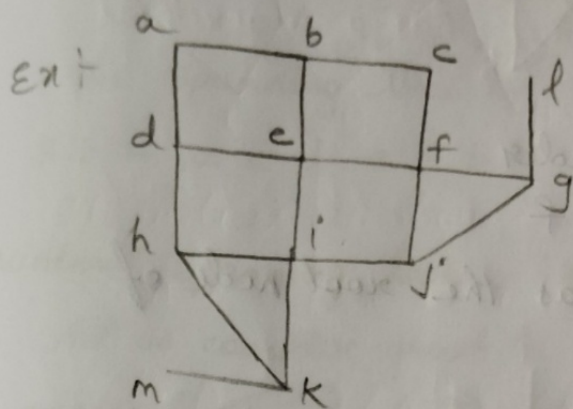
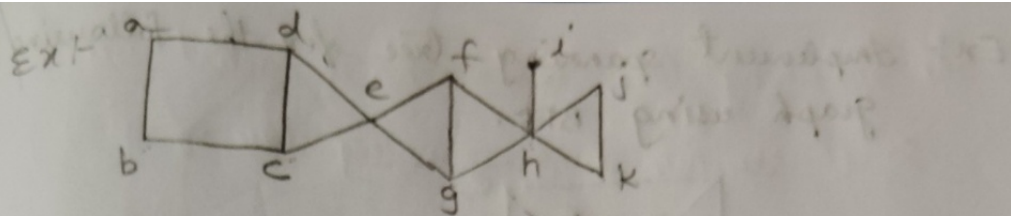
$$\{\underline{(C, F)}, (C, H)\}$$



$$\{(F, B), (F, C), \underline{(F, D)}\}$$



$$R.O = \{A, B, E, G, F, C, H, D\}$$



⇒ BFS Algorithm:

BFS is used to find the spanning tree T of given graph G .

Input: A connected graph G with n vertices
 $(v_1, v_2, \dots, v_n)$

Output: Spanning tree

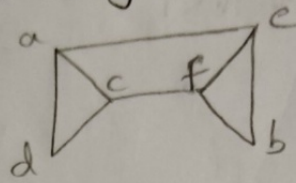
step i: let v_i be the root node of T .

step ii: Adding all adjacent edges of vertex v_i ,
 that or does not produce any cycle.

step iii: repeat step 2 until all vertices are visited.

After all vertices are ~~is~~ visited then stop.
 Then T is the spanning tree of given graph.

Ex:- Implement spanning tree for the following graph using BFS.



Sol:- All vertices in order
a b c d e f

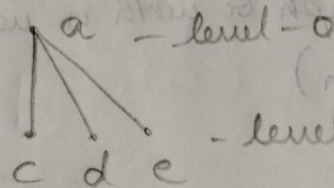
1. Now we select a as the root node of spanning tree T

a

2. Adjacent edges of a are

$\{(a,c), (a,d), (a,e)\}$

3. Then we add all edges to vertex 'a'



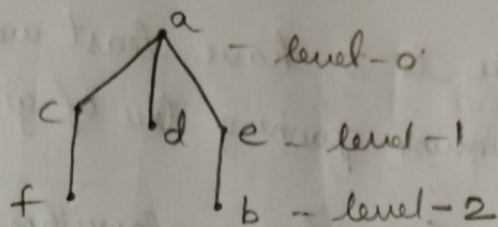
4. we need to find adjacent ^{edges} vertices of all vertices in level 1

$\{(c,a), (c,d), (c,f)\}$

$\{(d,c), (d,a)\}$

$\{(e,a), (e,b), (e,f)\}$

5. Add all adjacent edges that does not produce cycle.



The spanning tree of given graph is
 R.O = a c d e f b

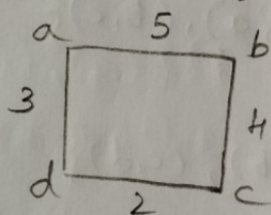
⇒ Minimal Spanning tree :- [MST] ^{weighted}

let us consider graph G is a ~~unweighted~~ ^{weighted} graph, a minimal spanning tree of graph G is a spanning tree of G with minimum weight.

(or)

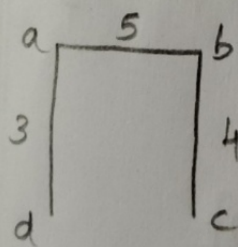
A spanning tree which has least weight is called a minimal spanning tree.

Ex:- Find the MST of given graph.



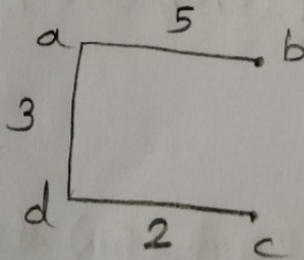
$$\begin{aligned}
 \text{Circuit rank} &\Rightarrow m - (n - 1) \quad , \quad m = 4 \\
 &= 4 - (4 - 1) \quad \quad \quad n = 4 \\
 &= 4 - 3 \\
 &= 1
 \end{aligned}$$

we have to remove 1 edge



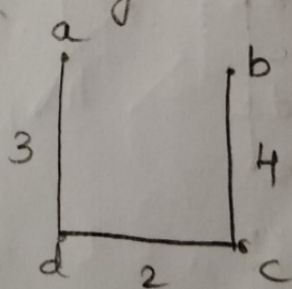
$$3 + 5 + 4 = 12$$

i)



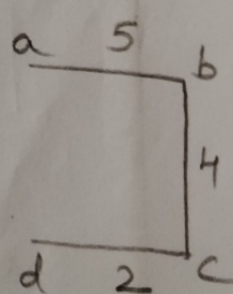
$$5 + 3 + 2 = 10$$

ii)



$$3 + 4 + 2 = 9$$

iii)



$$5 + 4 + 2 = 11$$

iv)

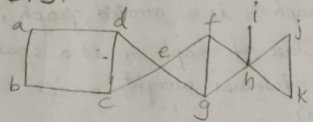
The spanning tree to 3 have least weight.
So it is minimum spanning tree of given graph.

Minimal Spanning tree has 2 algorithms.

1. Kruskal's

2. Prim's

DFS:



All vertices in graph are

a b c d e f g h i j k

* a $\{(a,b), (a,d)\}$

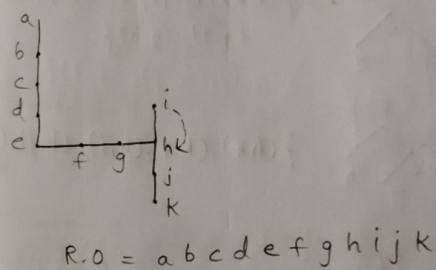
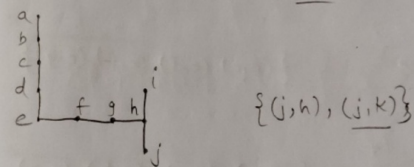
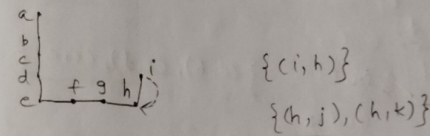
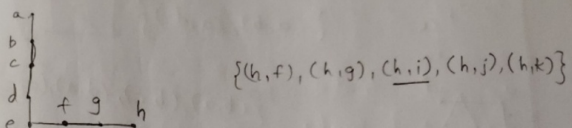
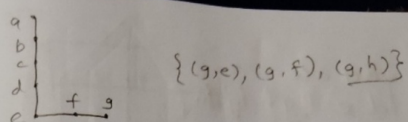
$\{(b,a), (b,c)\}$

$\{(c,b), (c,d)\}$

$\{(d,a), (d,e), (d,c)\}$

$\{(e,c), (e,d), (e,f), (e,g)\}$

$\{(f,g), (f,h), (f,e)\}$

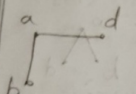


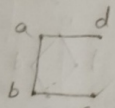
R.O = a b c d e f g h i j k

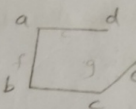
BFS:

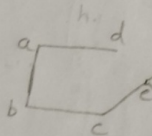
all vertices are
a b c d e f g h i j k

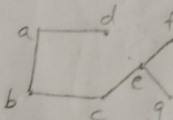
* a { (a,b), (a,d) }

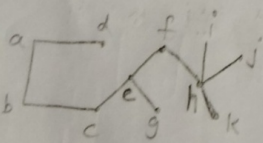
 { (b,a), (b,c), (d,a), (d,c) }

 { (c,b), (c,d), (c,e) }

 { (e,c), (e,d), (e,f), (e,g) }

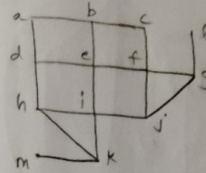
 { (f,e), (f,g), (f,h), (g,e), (g,f), (g,h) }

 { (h,f), (h,g), (h,i), (h,j), (h,k) }



R.O = a b d c e f g h i j k

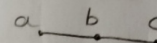
DFS :-

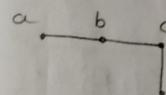


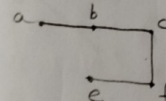
a b c d e f g h i j k l m

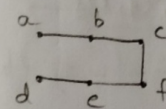
* a { (a,b), (a,d) }

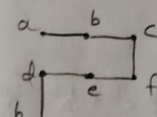
 { (b,a), (b,e), (b,c) }

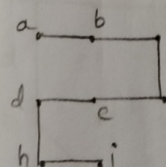
 { (c,b), (c,f) }

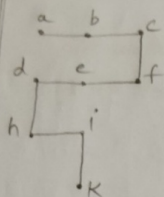
 { (f,c), (f,e), (f,g) }

 { (e,b), (e,d), (e,f), (e,i) }

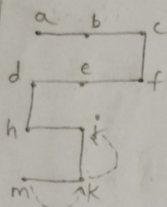
 { (d,a), (d,e), (d,h) }

 { (h,d), (h,i), (h,k) }

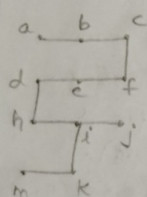
 { (i,h), (i,k), (i,j) }



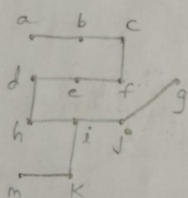
$\{(k, m)\}$



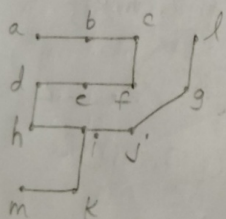
$\{(i, h), (i, k), \underline{(i, j)}\}$



$\{(j, f), (j, i), \underline{(j, g)}\}$



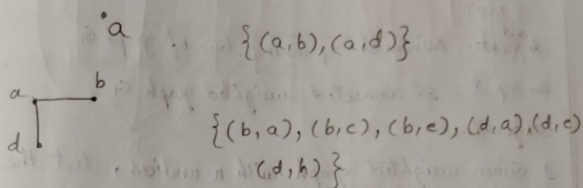
$\{(g, f), (g, j), \underline{(g, l)}\}$



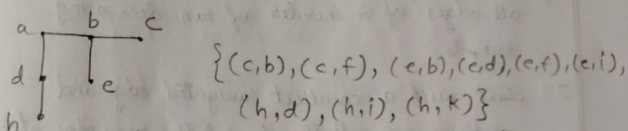
R.O = a b c f e d h i k m
j g l

BFS :

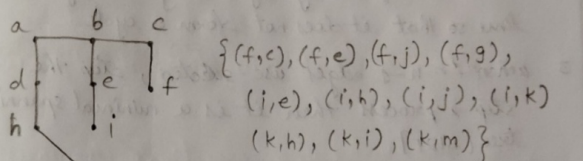
All vertices are a b c d e f g h i j k l m



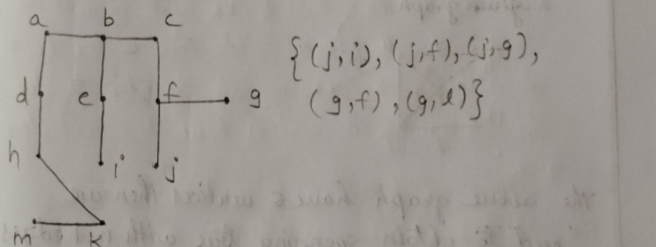
$\{(b, a), (b, c), (b, e), (d, a), (d, e), (d, h)\}$



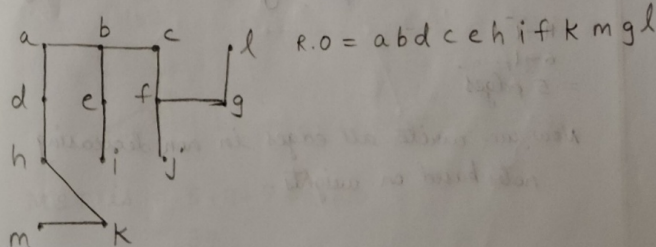
$\{(c, b), (c, f), (e, b), (e, d), (e, f), (e, i), (h, d), (h, i), (h, k)\}$



$\{(f, c), (f, e), (f, j), (f, g), (i, e), (i, h), (i, j), (i, k), (k, h), (k, i), (k, m)\}$



$\{(j, i), (j, f), (j, g), (g, f), (g, l)\}$



R.O = a b d c e h i f k m g l

6/1/24 Kruskal's Algorithm:-

It is used to obtain minimal spanning tree.

[MST]

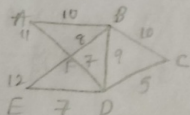
~~Input~~ ^{out}: minimal spanning tree of graph G .

~~Input~~: A connected weighted graph G .

Algorithm:-

1. Given weighted graph with n vertices, list the all edges of G in order of non decreasing weights.
2. start with a smallest weighted edge and continue sequentially by selecting one edge at a time so that it does not form a cycle.
3. when $n-1$ edges are selected, stop the step 2 process then it is a minimal spanning tree [MST]

Ex:- use Kruskal's algorithm find MST for given graph.



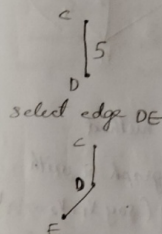
The above graph have 6 vertices then we need to obtain spanning tree with $n-1$ edges

$$\begin{aligned} n-1 \\ 6-1 \\ = 5 \text{ edges} \end{aligned}$$

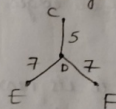
Now we write all edges in non decreasing order based on weights.

Ed	CD	DE	DF	BF	BD	AB	BC	AF	EF
WT	5	7	7	8	9	10	10	11	12

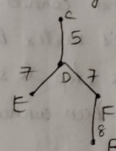
Now we select edge CD



Next select edge DF

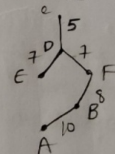


select edge BF



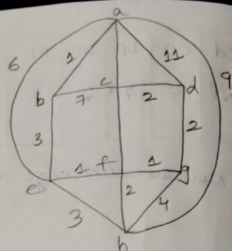
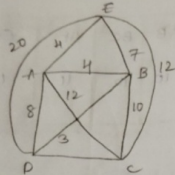
select BD already exists, BC, AF, EF

select AB



$$\begin{aligned} \text{MST is} &= 5 + 7 + 7 + 8 + 10 \\ &= 37 \end{aligned}$$

H.W

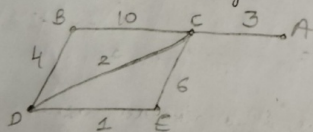


Prim's Algorithm:-

The working Rule for prim's method

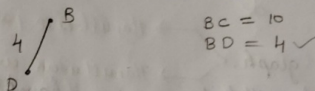
1. Given a connected weighted graph G with n vertices, assign n names (say $v_1, v_2, \dots, v_n$) or $(a, b, c, \dots)$. To these vertices and prepare a $n \times n$ in which weights of all edges are shown.
2. Start from vertex v_1 & connect to its nearest neighbour in V , now with minimal weight v_k . Now consider the edges v_1, v_k and connect to its closed neighbours. Let this vertex be BM from the vertex BM . Repeat the step to stop. The process when vertices have been connected $n-1$ edges.

ex:- using prim's algorithm



First we have to choose a vertex from the above graph B

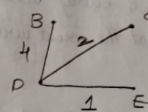
Now we have to choose & add the shortest edge from vertex B. There are 2 edges from vertex B that are BC & BD with 10 & 4 respectively among the edges the edge BD has the minimum weight so add it to the MST.



$$BC = 10$$

$$BD = 4 \checkmark$$

Next adjacent edge is D its adjacent edges are DB=4, DE=1, DC=2.



$$BC = 10$$

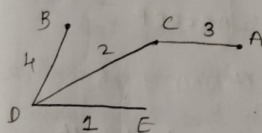
$$BD = 4 \checkmark$$

$$DB = 4 \checkmark$$

$$DE = 1 \checkmark$$

$$DC = 2 \checkmark$$

Next ~~B~~ C is working edge then adjacent are



$$B \rightarrow BC = 10$$

$$BD = 4 \checkmark$$

$$D \rightarrow CD = 2 \checkmark$$

$$DE = 1 \checkmark$$

$$BD = 4 \checkmark$$

$$C \rightarrow CA = 3 \checkmark$$

$$CB = 10 \checkmark$$

$$CD = 2 \checkmark$$

$$CE = 6 \checkmark$$

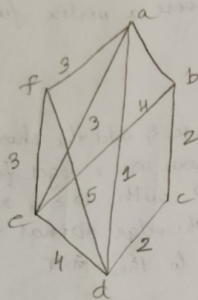
then select CB, already exists
then CE also exists.

then BC which is already exists.

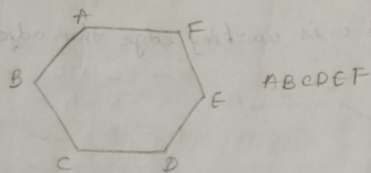
$$MST \text{ is } = 4 + 1 + 2 + 3$$

$$= 10$$

H/W.



⇒ Hamiltonian graph.
 → Hamiltonian path
 → Hamiltonian circuit.
 The graph will be known as a Hamiltonian graph if there is a closed walk in a connected graph, which passes each & every vertex of the graph exactly once except the start vertex or starting vertex.
 The Hamiltonian walk must not repeat any edge.

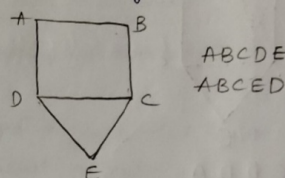


- In above graph there is a closed walk ABCDEF
- Except from the starting vertex, it passed through every vertex of graph exactly once.
- At the time of walk, the edges are not repeating.
- Due to all the reasons we can say that this graph is a Hamiltonian graph.

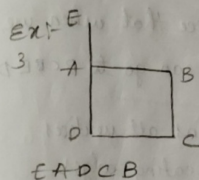
Hamiltonian path / Trail:

In a connected graph, if there is a walk that passes each & every vertex of a graph only once, this walk will be known as Hamiltonian walk, in this walk the edge should not be repeated.

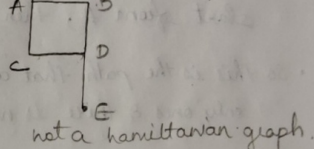
Ex:-



Ex:-



Ex:-

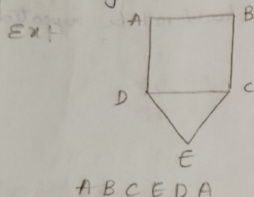


Ex:- We can see that there is no path that covers all the vertices only once. If we can start from F then we will go to ABC. After C we can either go to D or E, but we cannot go to both the vertices. That's why we can say that does not contain a Hamiltonian graph.

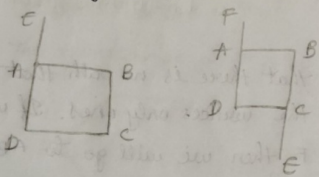
Hamiltonian Circuit.

In a connected graph if there is a walk if each & every vertex of the graph & after completing the walk, return to the starting walk where this type of graph is known as Hamiltonian.

- In H.C there must be no repeated edges, we must call it as hamiltonian circuit or cycle.



- In the above graph we can see that when we start from A, then we can go to B, C, E, D.
- So this is the path that contains all vertices only ones & there is no repeating edges.



This is not a Hamiltonian graph.

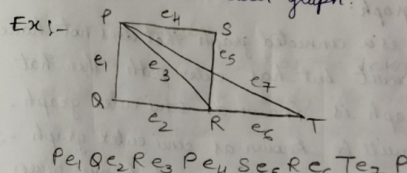
⇒ Eulerian Graph:

Euler path Euler circuit.

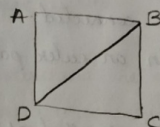
- Consider a connected graph G if there is a circuit C that contains all the edges of G , then that circuit is called an Euler circuit in G .

If there is a path in G , that contains all the edges of G then that is called Euler path in G .

- Recall that in a path & a circuit no edges can appear more than once but a vertex can appear more than once.
- This property is carried to Euler path & circuit also.
- A connected graph that contains an Euler circuit is called as Euler graph.



9/1/24 Euler circuit:

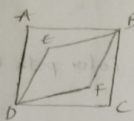


Note:- If all the vertices of graph contain the even degree then the type of graph is known as Euler circuit.

The above graph will contain the Euler circuit it starting the vertex and end vertex are same, & this graph visit each and every edge only once. The Euler circuit can contain the repeated vertex.

If we begin our path from vertex A and then go to vertices B, C, D & A. Then in this process the condition of same starting and end vertex is

satisfied but other condition covering all edge is not satisfied because there is one edge from vertex B to D. Hence the graph does not contain an Euler circuit.



It is a Euler circuit

Semi Euler graph:

If there is a connected graph that does not have a Euler circuit but has Euler path. Then that type of graph is known as semi Euler graph. Any graph will be known as semi Euler graph. If it satisfies two conditions which are given below.

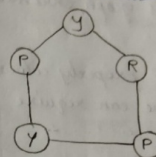
- for this graph must be connected
- this graph must contain an Euler path.

Graph colouring

graph colouring can be described as a process of assigning colours to the vertices of a graph in this the same colour should not be used to, till the two adjacent vertices we can also call graph colouring as vertex colouring.

In graph colouring we have to take care that a graph must not contain any edge whose end vertices are coloured by the same colour. This type of graph is known as properly coloured graph.

ex:- In this graph, we are showing properly coloured graph which are described as follows.



In above graph, the same colour cannot be used to the two adjacent vertices.

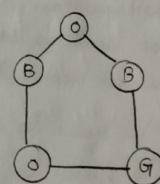
Hence we can call it as properly coloured graph

chromatic number

It can be described as the minimum no. of colours required in other word, the chromatic number can be described as a minimum number of colours that are needed to color in such a way that no two adjacent vertices of a graph will assign the same colour.

Ex:-

To understand the chromatic number we will consider a graph which is described as follows



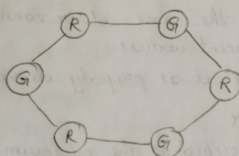
O - orange
B - Blue
G - Green

In the above graph,

- The same colour is not used to two adjacent vertices
- The minimum no. of colours of this graph is 3

which is needed to properly colour the vertices. In this graph chromatic number is 3.

ii) If we want to properly colour this graph in this case we can require atleast 3 colours.



chromatic number 3.

11/1/24

Unit-4

Elementary Combinatorics

There are two basic rules of counting

1. sum
2. product

1. sum: If a task, Suppose 2 task t_1 and t_2 are to be performed. The task t_1 can be performed in n different ways & t_2 can be performed in n different ways. And if 2 tasks cannot be performed simultaneously then one of the two tasks (t_1 or t_2) can be performed in $(n+n)$ ways.

- More generally if $t_1, t_2, t_3, \dots, t_k$ or k tasks such that no two of this tasks can be performed at the same time and if the task t_i can be performed n_i different ways. Then one of the k tasks ($t_1, t_2, t_3, \dots, t_k$) can be performed in $n_1 + n_2 + \dots + n_k$ ways.

Ex:- Suppose there are 16 boys & 18 girls in a class on mathematics. And we wish to select one of these subjects (either boy or girl) as the class representative.

Sol:- The no. of ways of selecting a boy is 16

The no. of ways of selecting a girl is 18

$\therefore$ The no. of ways of selecting a student (boy or girl) is $16 + 18 = 34$

Ex:- Suppose a library has 12 books on math, 10 on physics, 16 book on computer science and 11 books on electronics. Suppose students wishes to choose one book, which book he will choose.

Sol:-

The no. of math books = 12
 The no. of physics books = 10
 The no. of computer science = 16
 The no. of electronics book = 11

Total 49

⇒

Product Rule:

Suppose the two tasks t_1 & t_2 are to be performed one after the other. If t_1 can be performed in n_1 different ways.

If t_1 can be performed in n_1 different ways. And for each of these ways t_2 can be performed in n_2 different ways then both of the task can be performed in n_1, n_2 different ways.

More generally suppose that k tasks T_1, T_2, T_3 upto so on T_k are to performed in a sequence if T_1 can be performed in n_1 different ways and for each of these ways T_2 can be performed in n_2 different way. And for each of n_1, n_2 different ways of performing T_1, T_2 in that order T_3 can be performed in n_3 different ways and so on then the sequence of task $T_1, T_2, T_3, \dots, T_k$ can be

performed in $n_1, n_2, n_3, \dots, n_k$ different ways.

Ex:- Suppose a person has 3 shirts and 5 ties. no. of shirts = 3 how many ways.
 no. of ties = 5

Sol:-

total no. of ways = $3 \times 5 = 15$ ways.

Ex:- Suppose we wish to construct sequence of 4 symbols in which the first 2 are english letters, next 2 single digit number. If no letters or digit can be repeated then the no. of different sequence that we can construct is

Sol:- 1st English letter = 26

2nd English letter = 25

1st single digit number = 10

2nd single digit number = 9

Total ways = $26 \times 25 \times 10 \times 9$
 = 58,500

with repetition = $26 \times 26 \times 10 \times 10$
 = 67,600

Ex:- Suppose a restaurant sales 6 south Indian dishes, 4 north India, 3 hot beverages & 2 cold beverages. For breakfast a student wishes to buy 1 south Indian and 1 ^(parotta) ^(hot) beverage or 1 north Indian and 1 cold beverage.

1 SI & 1 hot beverage = $6 \times 3 = 18$ ways

1 NI & 1 cold beverage = $4 \times 2 = 8$ ways

Total sales = $18 + 8 = 26$ ways.

Sol:-

1. In how many ways can we distribute 10 identical marbles among 6 distinct containers?

Sol: no of marbles = 10
no of containers = 6
no of ways = $\frac{10+6-1}{6-1} = \frac{15}{5} = 3$

2. From a group of 7 men and 6 women, 5 persons are to be selected to form a committee so that at least 3 men are on the committee, in how many ways it can be done.

ways:- 3 men + 2 women
7 men + 1 woman
5 men + 0 woman

$$\begin{aligned} \text{No. of ways} &= {}^7C_3 \times {}^6C_2 + {}^7C_4 \times {}^6C_1 + {}^7C_5 \times {}^6C_0 \\ &= \frac{7!}{(3! \times 4!)} \times \frac{6!}{(2! \times 4!)} + \frac{7!}{(4! \times 3!)} \times \frac{6!}{(1! \times 5!)} + \frac{7!}{(5! \times 2!)} \times \frac{6!}{(6! \times 0!)} \\ &= 756 \end{aligned}$$

3. How many committees of 4 with a given chair person can be selected from 12 person?

A chair person can be selected in 12 ways.

Now, we need to select 3 person out of remaining 11.

$\therefore$ No of committees of 4 person with a chair person can be selected in $12 \times {}^{11}C_3$ ways

$$\begin{aligned} \text{i.e. } 12 \times \frac{11!}{3! \cdot 8!} &= \frac{12 \times 11 \times 10 \times 9 \times 8!}{3! \cdot 8!} \\ (or) {}^{12}C_4 &= \frac{12!}{4! \cdot (12-4)!} = \frac{11880}{24} = 495 \end{aligned}$$

4. In how many ways we can distribute 12 identical pencils to 5 children so that every child gets at least one pencil?

Formula = $(n-1) \text{ choose } (x-1)$
 n = total no of items, x = no of groups
12 pencils to 5 children

Take $(12-1) \text{ choose } (5-1)$

$${}^{11}C_4 = \frac{11 \times 10 \times 9 \times 8}{4!} = 330$$

5. a) how many ways are there to sit 10 boy & 10 girls around a circular table

b) how many ways are there 3 persons sit around a round table.

Sol: a) let one of them be seated anywhere. Then the remaining $n-1$ persons can be seated in $(n-1)!$ ways. This is total no. of ways of arranging the n persons in a circle.

$$\text{total} = 20$$

1 seated anywhere

$$\text{so remaining } (20-1) = 19$$

$$(n-1)! = (19-1)! = 18!$$

b) 1 person anywhere, total = 3

$$(2-1)! = 1! = 1$$

6. Find the no of permutations of the letters of the word ENGINEERING.

Sol: The given word is ENGINEERING has 11 letters out of these letters $E=3$, $N=3$, $G=2$, $R=1$, $I=2$

The total no of permutations of the letters of word ENGINEERING is = $\frac{n!}{n_1! n_2! n_3! n_4! n_5!}$

2,70,200 ways.

$$= \frac{11!}{3! \times 3! \times 2! \times 1! \times 2!}$$