Chapter 5

Introduction to graph theory

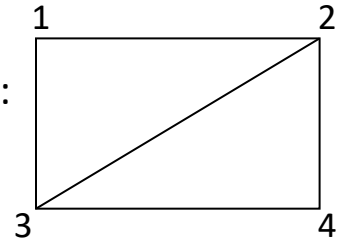
5.1. **Graph**

A graph G consists a finite set of vertices (nodes) V and a finite set of edges E.

i.e, $G = (V, E)$, where $E = \{\{v_i, v_j\} : v_i, v_j \in V\}$

Example:- $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ is undirected edges.

Hence the undirected graph $G = (V, E)$ becomes G:



- 5

Note:- i. An edge $\{a, b\}$ stands for $\{(a, b), (b, a)\}$

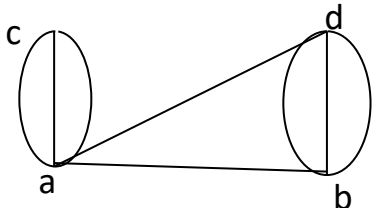
ii. $(a, b) = (b, a)$ only when $a = b$

5.1.1 **Order and size**

The number of vertices in a graph $G(V, E)$ is called its order, denoted by $|V|$

The number of edges in a graph $G(V, E)$ is called its size, denoted by $|E|$

e.g. Let G:

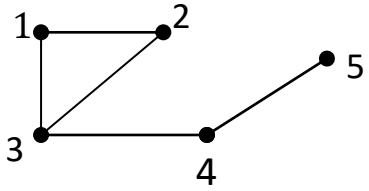


Then order of G or $|V|$ =4 and size of G or $|E|$ = 8

5.1.2 **Adjacent vertices**

Two vertices v_i, v_j are adjacencies if there exits an edge $\{v_i, v_j\}$ in G

e.g. Let G:



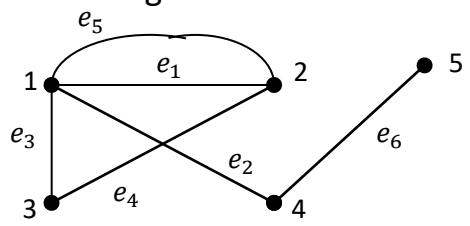
Then 1&2 are adjacent vertices, 1&3 are adjacent vertices,

2&3 are adjacent vertices, 4&5 are adjacent vertices

5.1.3 **Parallel edges**

If there is more than one edge between the same pair of vertices, then the edges are said to be parallel edges.

e.g. Let G:

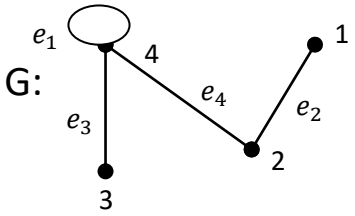


Then e_1 and e_5 are parallel edges.

5.1.4. **Loop**

It is an edge whose starting and ending vertex are the same.

e.g. Let G:



Then e_1 is a loop.

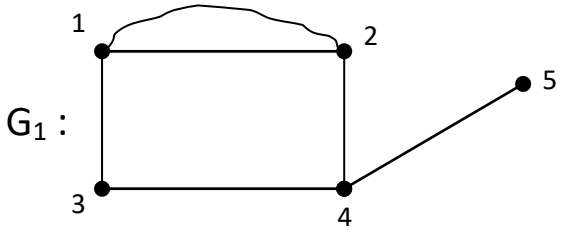
Note :- A graph G has no loop is called loop – free.

5.2. **Kinds of graph**

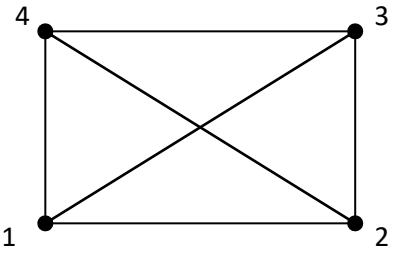
5.2.1. **Simple graph:-**

It is a graph $G(V, E)$ that has no self loop or parallel edges.

e.g. Let G_1 :



G_2 :



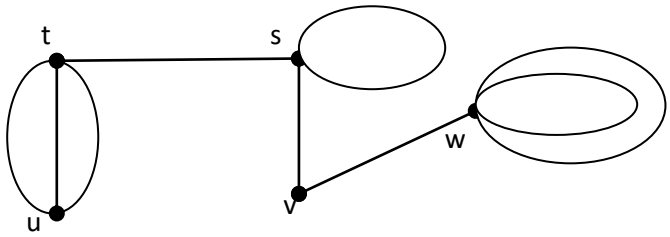
Then G_1 is not a simple graph but G_2 is a simple graph.

5.2.2. **Multi graph**:- if it contains parallel edges. e.g. G_1 is a multi graph

5.2.3. **pseudo graph**:- if it contains both parallel edges & loops.

e.g

G :



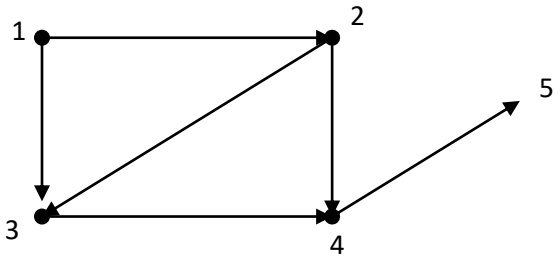
5.3. **Digraph (Directed graph)**

It is a graph (V, E) , where V is a set of nodes or vertices and E is the set of edges having direction.

Note:- If $(v_i,\ v_j)$ is an edge, then there is an edge from $v_i\ to\ v_j$.

e.g. Let $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (4, 5)\}$. Then the digraph G becomes

G:



and also i) 1 is said to be adjacent to 2

ii) 2 is said to be adjacent from 1

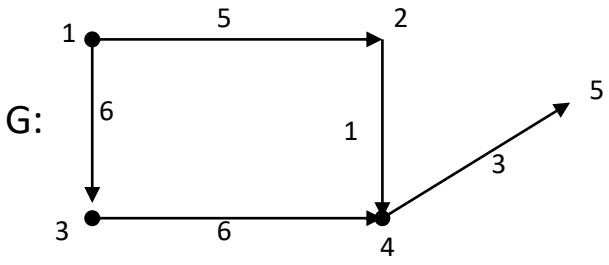
iii) 1 is called origin or source & 2 is terminus or terminating vertex.

5.4. **Weighted graph**:-

A graph (or digraph) is known as a weighted graph (or digraph) if each edge of the graph has some weights.

Consider $V = \{1, 2, 3, 4, 5\}\ and\ \ E = \{e_1,\ e_2,\ e_3,\ e_4,\ e_5\}$, where $e_1 = (1, 2)$, $e_2 = (1, 3)$, $e_3 = (2, 4)$, $e_4 = (3, 4)$, $e_5 = (4, 5)$ and $w(e_1) = 5$, $w(e_2) = 6$,

$w(e_3) = 1$, $w(e_4) = 6$, $w(e_5) = 3$. Hence the weighted graph G becomes:



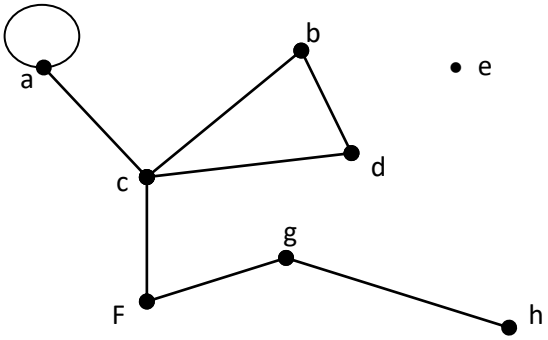
5.5. **Degree of a vertex**

- ✓ The number of edges connected to the vertex v is called degree of vertex v, denoted degree (v) or deg (v)
- ✓ In case of a diagraph, there are two degree, i.e. indegree & outdegree.
- ✓ The number of edges coming to the vertex v is known as indegree of v.
- ✓ The number of edges emanating from the vertex v is known as outdegree of v.

Note:- In case of loop, degree of a vertex is 2

e.g.

Let G:



Then deg(b) = deg(d) = deg(f) = Deg(g) = 2, deg(c)=4, deg(e) = 0, deg(h)= 1, deg(a)=3.

5.5.1. **Isolated vertex**

A vertex is said to be an isolated vertex if there is no edge connected from any other vertex to that vertex.

In other words a vertex is said to be an isolated vertex if the degree of that vertex is zero.

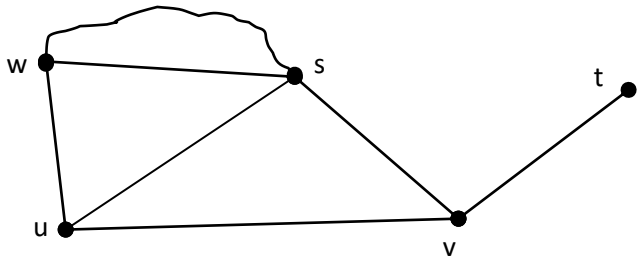
e.g in the above example “e” is isolated vertex.

5.6. Path

A path in a graph is a sequence $v_1,\ v_2,\ v_3, - - - , v_k$ of vertices each adjacent to the next, and a choice of an edge between each v_i to v_{i+1} so that no vertices is chosen more than once.

e.g. 1 .

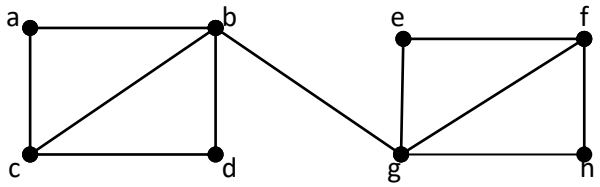
Let G:



Then one path is wsuvt, 2nd path is wusvt

2.

Let G:

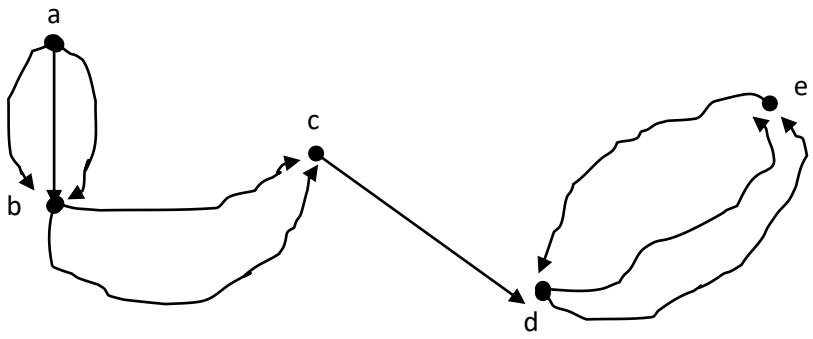


Then i) How many paths are there in G from a to h? 9

ii) How many paths having length 5 from a to h? 3

3.

Let G:



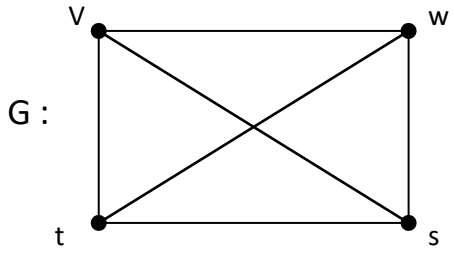
Then

- G is directed multigraph.
- There are 3 edge from a to b.
- Edge (a, b) has multiplicity 3.
- Edge (b, c) and (d, e) have multiplicity 2.
- Edge (e, d) and (c, d) have multiplicity 1.

5.7. Cycle

If there is a path containing one or more edges which starts from a vertex v and terminates into the same vertex, then the path is known as a cycle.

e.g. Let



Then one cycle is vwsv, an other cycle are vstv, vwstv.

5.8. Matrix representation of graphs

A. Adjacency matrix

The general form of adjacency matrix is: A = $\begin{bmatrix} a_{11} & a_2 & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ . & & & & & \\ . & & & & & \\ . & & & & & \\ a_{n1} & a_{n2} & . & . & . & a_{nn} \end{bmatrix}$

Where $a_{ij} = \begin{cases} 1; if\ there\ is\ an\ edge\ from\ v_i\ to\ v_j \\ 0, \ otherwise \end{cases}$

This is also known as bit matrix or Boolean matrix as each entry is either 1 or 0.

Note:- 1. In the **adjacency matrix**,

if the graph is a simple graph, then the main diagonal elements are zero.

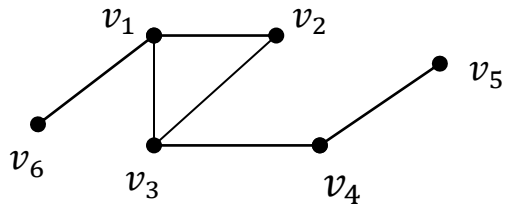
2. In a **multi graph**, the adjacency matrix can be found out with the relation

$$a_{ij} = \begin{cases} n; n\ be\ the\ number\ of\ edges\ from\ v_i\ to\ v_j \\ 0, \ otherwise \end{cases}$$

3. In a weighted graph, the adjacency matrix can be found out with the relation

$$a_{ij} = \begin{cases} w; w\ is\ the\ weight\ of\ the\ edges\ from\ v_i\ to\ v_j \\ 0, \ otherwise \end{cases}$$

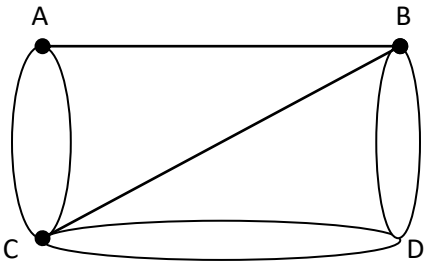
e.g 1. Let a graph

G: 

Then the adjacency matrix is given as:

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

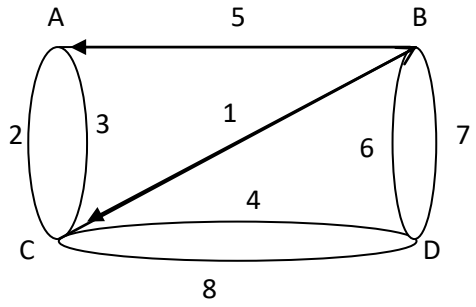
e.g. Let

G: 

Then the adjacency matrix of the graph with respect to ordering A, B, C, and D is

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix} \end{array}$$

e.g. 3. Let

G: 

Then the adjacency matrix of the graph w.r.t the ordering A, B, C and D is

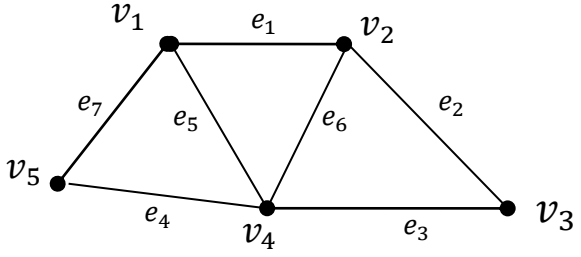
$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 0 & 3 & 0 \\ 5 & 0 & 1 & 7 \\ 2 & 0 & 0 & 4 \\ 0 & 6 & 8 & 0 \end{bmatrix} \end{array}$$

B. Incidence matrix

Suppose G be a simple undirected with 'm' vertices and 'n' edges, then the incidence matrix is a matrix of order (mxn), where the element a_{ij} defined as

$$a_{ij} = \begin{cases} 1; if\ vertex\ v_i\ belongs\ to\ edges\ e_j \\ 0,\ otherwise \end{cases}$$

e.g. Let the graph

G: 

Then the incidence matrix of G is of order (5x7) the incidence matrix relative to the ordering $v_1,\ v_2,\ v_3,\ v_4,\ v_5$ and $e_1, e_2, e_3, e_4, e_5, e_6, e_7$ is :

$$\mathbf{I} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

5.9. **Connected graph**:-

A graph G is said to be connected if there is a path for every pair of distinct vertices 'u' and 'v' in G.

Definition:-

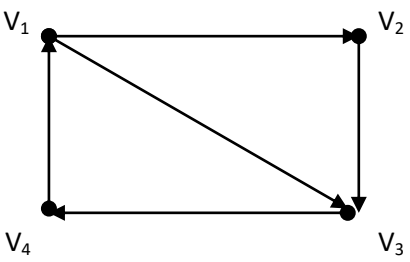
i) A directed graph is said to be strongly connected if for every distinct vertices 'u' and 'v' in G, there is a directed path from 'u' to 'v' and also from 'v' to 'u'.

ii) A directed graph is said to be weakly connected if for every distinct vertices in G, there is a path without taking the direction.

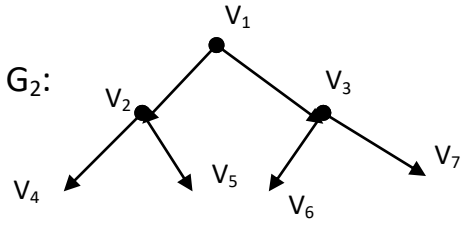
iii) A directed graph is said to be connected if its associated undirected graph (obtained by ignoring the directions of the edges) is connected.

e.g. Let

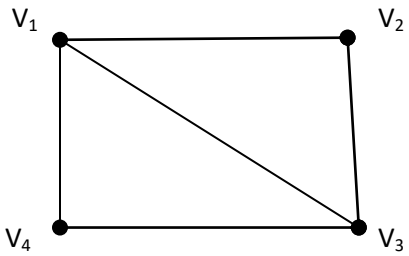
G_1 :



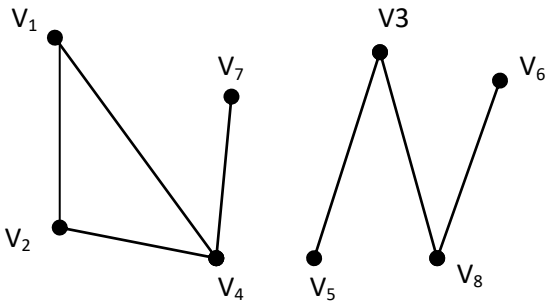
G_2:



G_3 :



G_4 :



Then G_1 is strongly connected.

G_2 is weakly connected

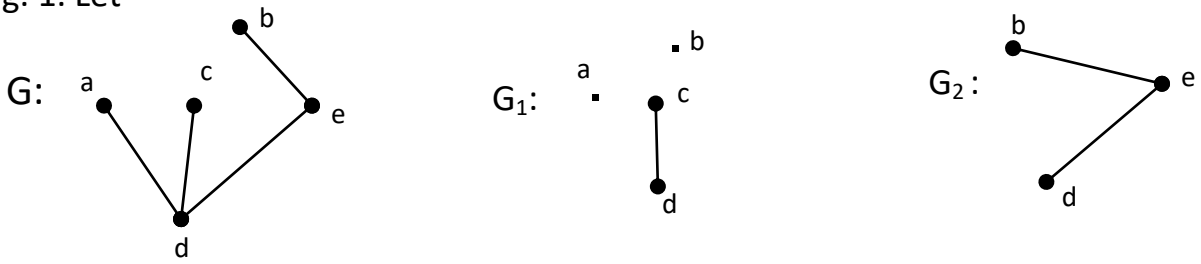
G_3 is connected

G_4 is disconnected

5.10. Sub graph:-

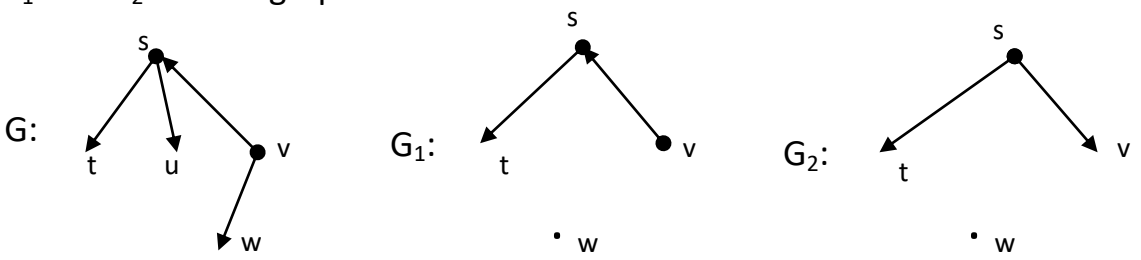
If $G = (V,\ E)$ is a graph (digraph), then $G_1 = (V_1,\ E_1)$ is called a sub graph of G if $\emptyset \neq V_1 \subseteq V$ and $E_1 \subseteq E$, where each edge in E_1 is incident with vertices in V_1.

e.g. 1. Let



Then G_1 and G_2 are subgraph of G.

2. Let



Then G_1 is subgraph of G

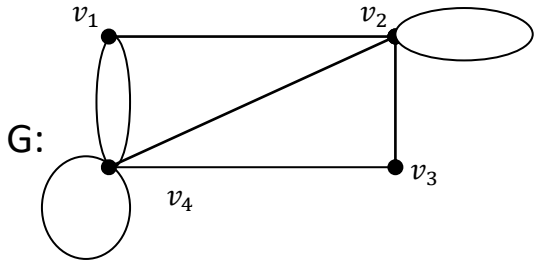
G_2 is not subgraph of G because(s, v) is a subset of E_2 but (s, v) is not a subset of E.

5.10.1. Vertex deleted subgraph:

Suppose $G = (V,\ E)$ be graph. If we delete a subset U of V and all the edges, which have a vertex in U as an end, then the resultant graph is termed as vertex deleted subgraph of G.

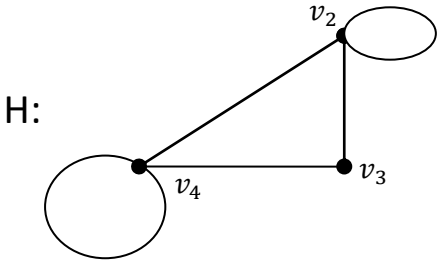
e.g

Let



Then by deleting the vertex v_1 ,

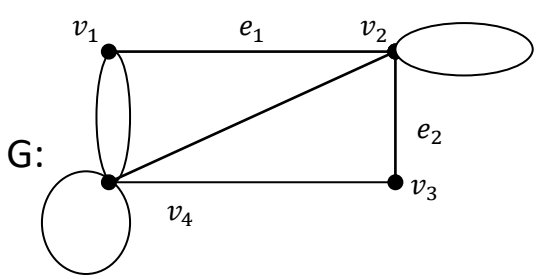
the vertex deleted subgraph H is



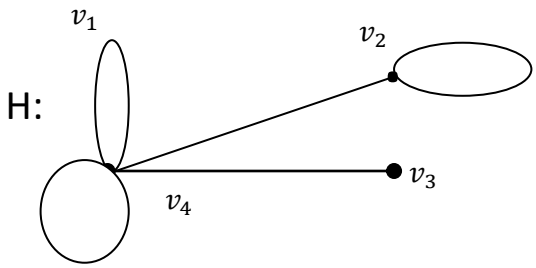
5.10.2. Edge deleted subgraph

Suppose $G = (V,\ E)$ be a graph. If a subset F from the set of edges E is deleted from the graph G, then the resultant graph is edge deleted subgraph of G.

e.g Let



deleting e_1 and e_2, then the edge deleted subgraph is



5.11 Walks:-

Let x, y be (not necessary distinct) vertices in an undirected graph $G = (V,\ E)$.

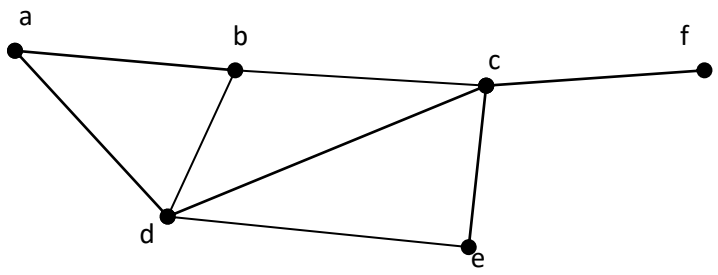
An $x - y$ walk in G is (loop – free) finite alternating sequence

$x = x_0,\ e_1,\ x_1,\ e_2, - - - , x_{n-1},\ e_n,\ x_n = y$ of vertices and edges from G, starting a vertex 'x' and ending at vertex 'y' and involving the 'n' edges $e_i = \{x_{i-1}, x_i\}$, $1 \leq i \leq n$

1. The length of this walk is n , the number of edges in the walk.(when $n = 0$, there are no edges, $x = y$ and the walk is called trivial)
2. Any $x - y$ walk, where $x = y\ (n > 1)$ is called closed walk. Otherwise the walk is open.
3. If no vertex of the $x - y$ walk occurs more than once, then the walk is called an $x - y$ path.

e.g

Let G:



1. $\{a,\ b\}, \{b,\ d\}, \{d,\ c\},\ \{c,\ e\},\ \{e,\ d\},\ \{d,\ b\}$ This is $a - b$ walk of length 6.
2. $b \to c \to d \to e \to c \to f$: in this case $b - f$ walk has length 5 and c is repeated, but no edge appears more than once.

3. $\{f,\ c\},\{c,\ e\},\{e,\ d\},\ \{d,\ a\}$: in this case $f-a$ walk has length 4 with no repetition of either vertices or edges.

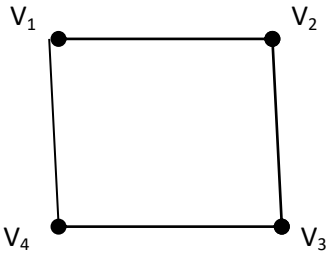
5.12 Operations on graphs

A. Union

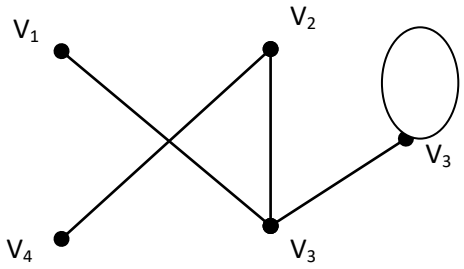
If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, then the union $(G_1 \cup G_2) = (V, E)$ is a graph with $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$

e.g Find the union of the following graphs

G_1 :



G_2 :

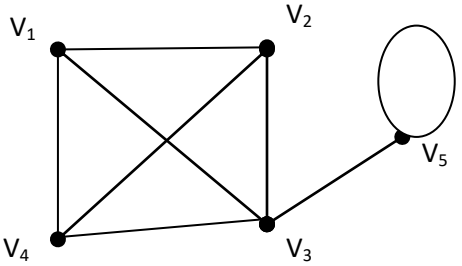


Solution: $V = \{v_1, v_2, v_3, v_4, v_5\}$,

$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_1, v_3\}, \{v_2, v_4\}, \{v_5, v_5\}, \{v_3, v_5\}\}$

Hence the graph $G_1 \cup G_2$ becomes

$G_1 \cup G_2$:

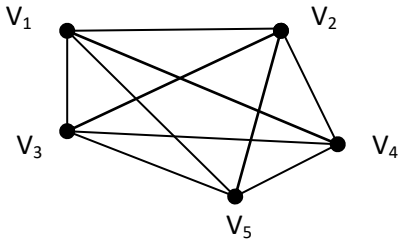


B. Intersection

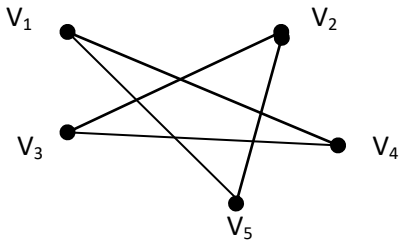
If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with at least one vertex in common, then their intersecton $(G_1 \cap G_2) = (V, E)$ is a graph with $V = V_1 \cap V_2$ and $E = E_1 \cap E_2$.

e.g Find the intersection of the following graph

G_1:



G_2:

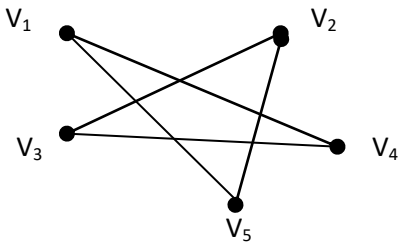


$V_1 = \{V_1, V_2, V_3, V4, V_5\}$ and $V_2 = \{V_1, V_2, V_3, V4, V_5\}$,

Then $V = \{V_1, V_2, V_3, V4, V_5\}$ and $E = \{\{v_1,\ v_4\},\ \{v_1, v_5\}, \{v_2, v_3\}, \{v_2, v_5\}, \{v_3, v_4\}\}$

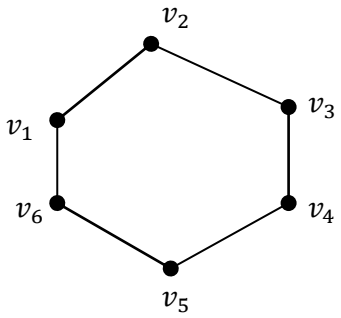
Thus the graph $G_1 \cap G_2$ becomes:

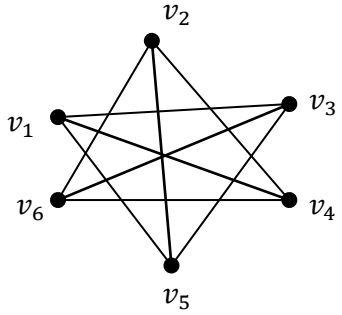
G_2:



C. Complement

Suppose G be simple graph with n – vertices. Then the complement of G is given by $\bar{G}$ and is defined to the simple graph with the same vertices G and where vertices (u, v) are adjacent in $\bar{G}$ if u and v are not adjacent in G.

e.g Let G: 

, then $\bar{G}$: 

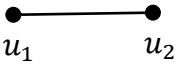
D. Product of graphs

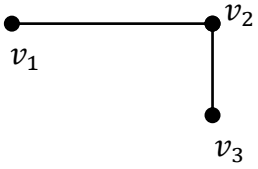
Suppose $G_1 = (V_1,\ E_1)\ and\ G_2 = (V_2,\ E_2)$ be graphs. The product of $G_1\ and\ G_2$, denoted $G_1 \times G_2$ is defined as $G_1 \times G_2 = (V,\ E)$, where $V = V_1 \times V_2$ and the edge set E can be found out from the following relation:

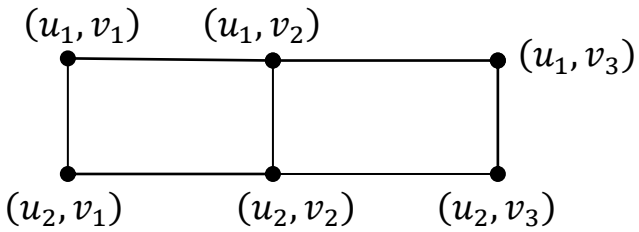
If $(u_1,\ u_2)$ and $(v_1,\ v_2)$ be two vertices of $G_1 \times G_2$, then there is an edge between them When i) ($u_1 = v_1$ and u_2 is adjacent to v_2) or

ii) ($u_2 = v_2$ and u_1 is adjacent to v_1)

e.g Consider the graphs G_1 and G_2 as:

G_1: 

G_2: 

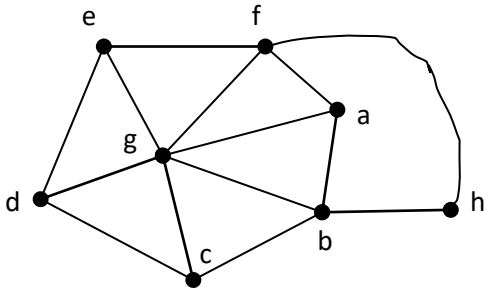
Then $G_1 \times G_2$: 

5.13 Graph coloring and chromatic polynomials

Definition:- The digraph $G = (V,\ E)$ is said to be coloring if adjacent vertices have different colors.

The minimum number of colors needed to proper color G is called the chromatic number of g and is written as $\mathcal{X}(G)$.

e.g

Let G: 

Then $\mathcal{X}(G) = 3$.

Let λ be the number of color. Then $P(G,\lambda)$, in the variable λ is called **the chromatic polynomial** of G, that will tell us in how many different ways we can properly color the vertices of G, using at most λ colors.

5.14. Trees

Definition:- A tree is a non – directional graph is connected and does not have a circuit.

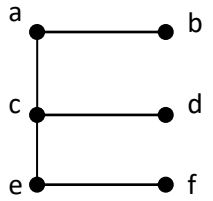
Or A tree is connected graph that i) has no cycles

ii) has no loopes

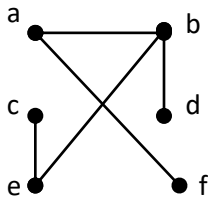
iii)has no multiple edges.

e.g

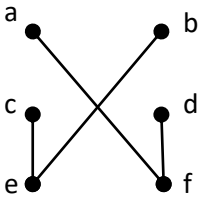
G_1:



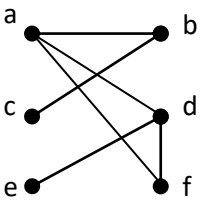
G_2:



G_4:



G_3:



Then G_1 and G_2 are tree. But G_3 and G_4 are not tree. Because G_4 is not connected.

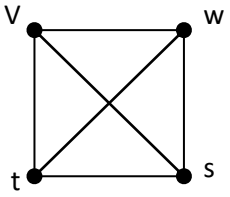
Types of trees

1. **Spanning tree**

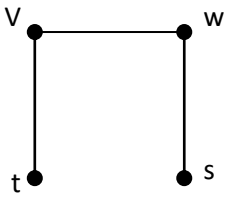
   It is a subgraph of a given graph G and is obtained by eliminating the circuit in a graph and it contains all vertices of G.

   e.g Consider the graph

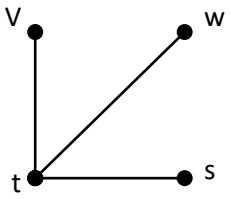
   G :



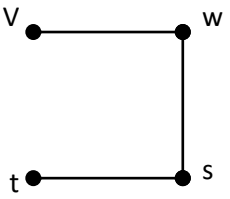
   Then the spanning tree may be



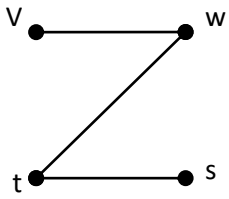
or



or



or



2. Rooted Tree

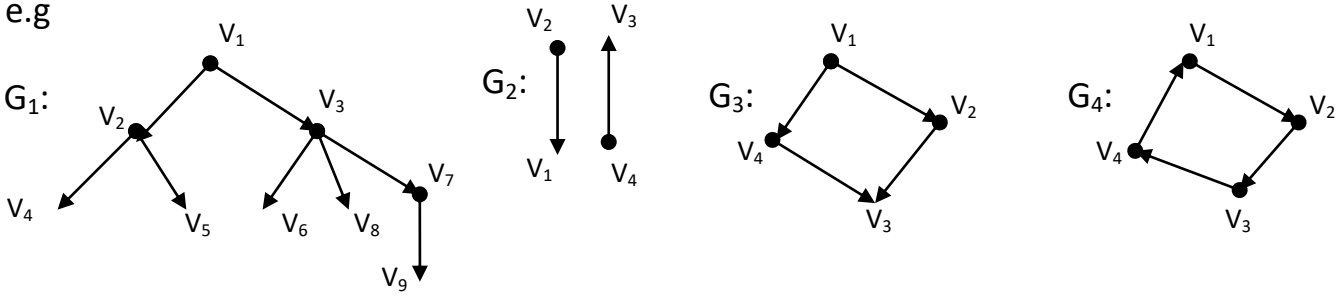
A tree is a digraph with a non empty set of nodes such that

i) there is exactly one node, called the root of the tree, which has indegree 0.

ii) Every node other than the root has indegree 1.

iii) For every node 'a' of the tree, there is a directed path from the root to 'a'.

e.g



Then G_1 is tree but G_2, G_3, G_4 are not tree.

In G_1 ,- V_1 is called root,

- V_2 , V_3 & V_7 are internal nodes (vertices).
- V_4 , V_5 , V_6, V_8 & V_9 are leaves, which do not have children
- V_1 is a father (parent) of V_2 & V_3, that is, V_2 , V_3 are the son of V_1
- V_3 is a father (parent) of V_6, V_8 & V_7
- V_1 is ancester of V_9
- In a horizontal level, V_1 is on level '0'; V_2 & V_3 are on level '1'; V_4 , V_5, V_6 , V_8 & V_7 are on level '2'

Note: Length of the longest path to a leaf from the root is called height (depth) of the tree.

Binary tree:- Each nodes has 2, 1, or 0 sons, i.e, no nodes has more than 2 sons.

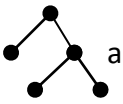
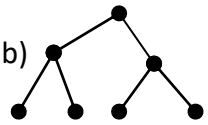
Ternary tree:- Each nodes has 3, 2, 1, or 0 sons, i.e, no nodes has more than 3 sons.

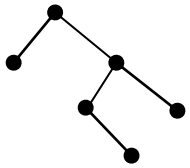
n-ary tree:- Each node has ' I ' sons, $0 \leq I \leq n$.

Complete n- ary tree:- Each node has '0' or 'n' sons.

Complete binary tree:- Each node has '0' or '2 ' sons.

Example:-

a)  and b)  are complete binary tree.

but c)  not complete binary tree, it is binary tree

Traversal Methods:-

Tree traversal algorism (or there are 3 methods in which we can traversal nodes)

Pre order:- 1. Process the root node r of tree (T)

2. if T_1 (left sub tree) exists, then process T_1

3. if T_2 (right sub tree) exists, then process T_2 (i.e, root, left, right)

In order :- 1. if T_1 (left sub tree) exists, then process T_1

2.Process the root node r of tree (T)

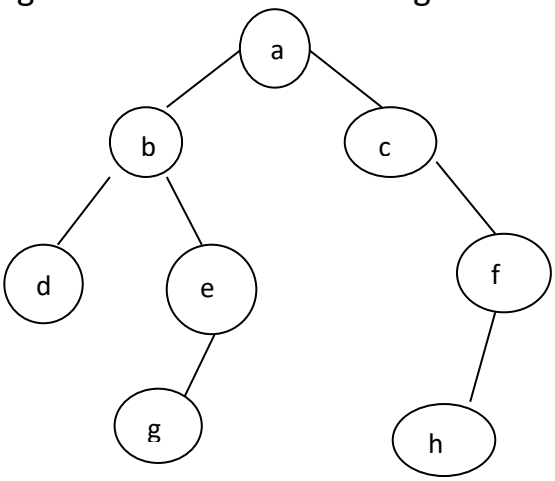
3. if T_2 (right sub tree) exists, then process T_2 (i.e, left, root, right)

Post order:- 1. . if T_1 (left sub tree) exists, then process T_1

2. if T_2 (right sub tree) exists, then process T_2

3. Process the root node r of tree (T) (i.e, left, right, root)

e.g. 1. Consider the following tree:



Then 1. Pre order:- a b d e g c f h

2. In order:- d b g e a c h f

3. Post order:- d g e b h f c a

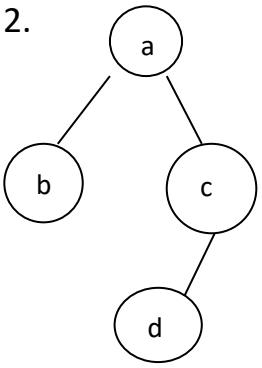
Ex. 2. Let pre order is abcd and in order is badc. Then draw the tree.

3. Let pre order is abcd and post order is dcba. Then draw the tree.

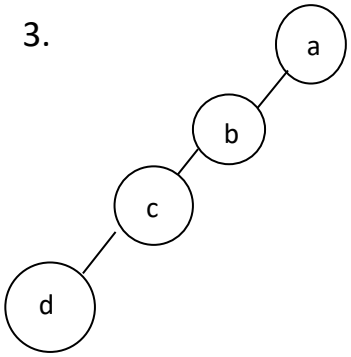
4. Let pre order is abdehkcfigJl and in order is dbhkeaficglJ

Solution:-

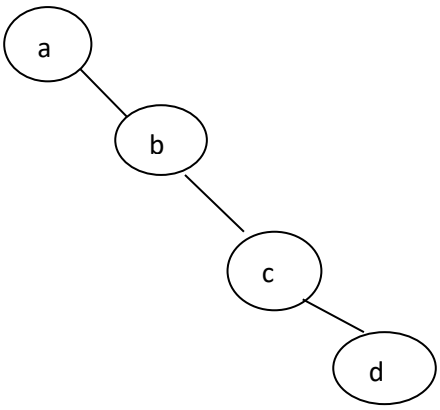
2.



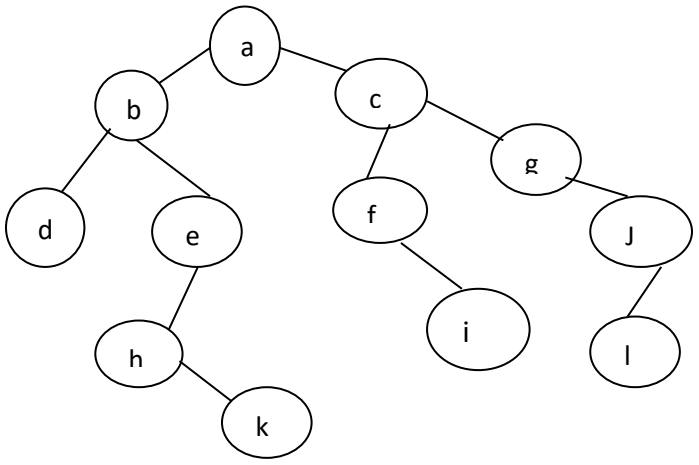
3.



or



4.



Minimum spanning tree (MST) is a subset of edges that i) reaches every vertex.

ii) has minimum total weight

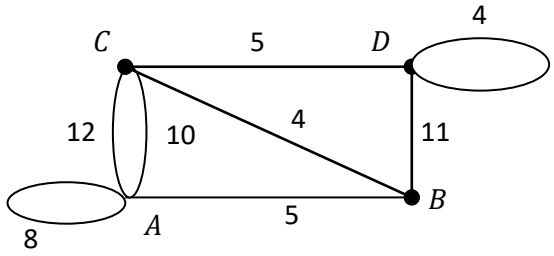
iii) has no cycles

To draw the tree, use the following steps:

1. Remove all the loops if exists.
2. Remove all parallel edges between two vertices except the one with least weight.
3. Select the sides of the minimum weighted graph G, enter in to T
4. Select the edge $(u,\ v)$ in G which has a minimum weight and side by side with vertices in T, which provide that side does not form a circuit in T.
5. Repeat step 2 for n-2 times.

e.g Find MST if the following graph is given.

G:

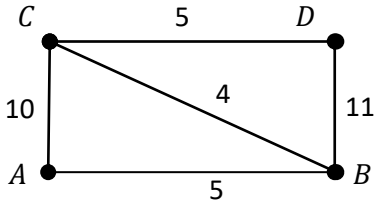


Solution:- step1:- Remove all the loops

Step2:- Remove all the parallel edges

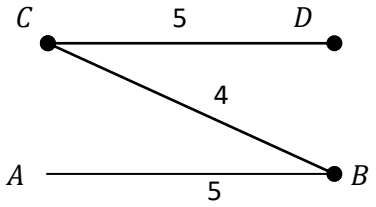
Now, the graph becomes,

G:



- Start from vertex A. Find the smallest value in a A – row except ‘0’ (i.e, 5, so we mark edge AB) and find the smallest value in a B – row & A – row (i.e, 4 is the smallest unmarked value in A – row & B – row, so we mark edge BC)
- Then find the smallest value in A – row, B – row, C – row. (i.e, 5 is the smallest un marked value in A – row, B – row, C – row, so mark CD).
- Finally, Draw MST,

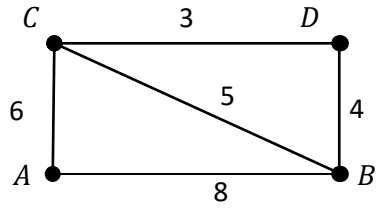
G:



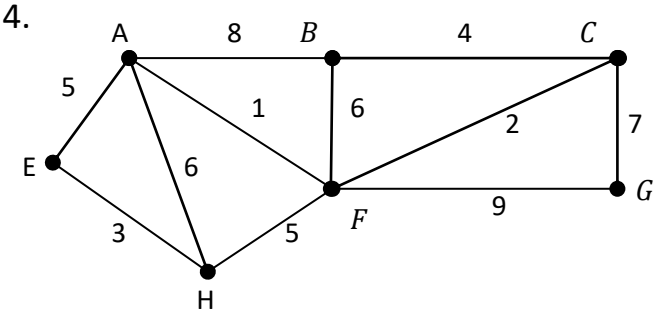
Exercise:- Find MST for

1.

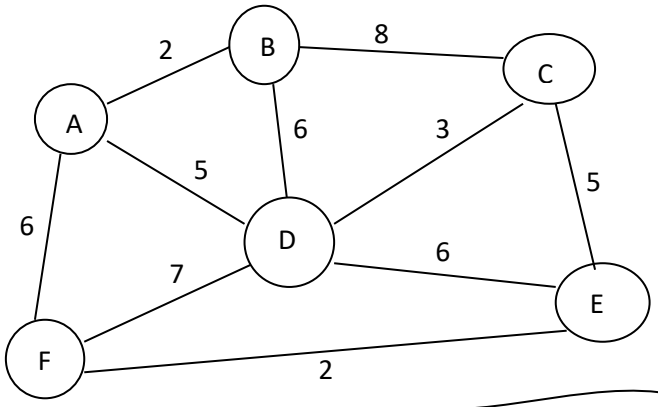
G:



4.



2.



3.

