Graph Theory

Graph:

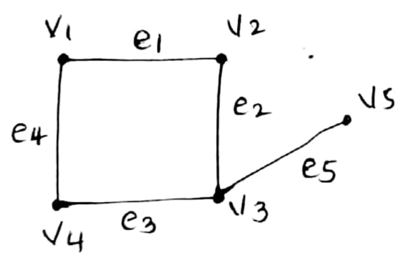
A graph G is a pair of sets (V, E) where, V is a set of vertices and E is a set of edges.

The most common representation of a graph is a diagram with vertices and edges.

(1) The vertices (or) nodes are represented as points or small circles.

(2) Edges are represented as line segments (or) curve joining of its end vertices.

eg:



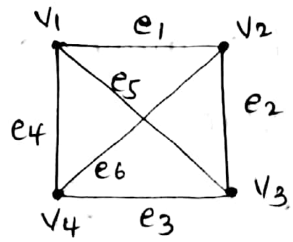


fig: Representation of a graph with vertices and edges

→ The first graph consists of 5 vertices and 5 edges so,

$V = \{v_1, v_2, v_3, v_4, v_5\}$ and $E = \{e_1, e_2, e_3, e_4, e_5\}$.

→ In the second graph, there are 4 vertices and 6 edges. So,

$V = \{v_1, v_2, v_3, v_4\}$ and $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$

There are 2 types of graphs. They are

(1) Directed graph

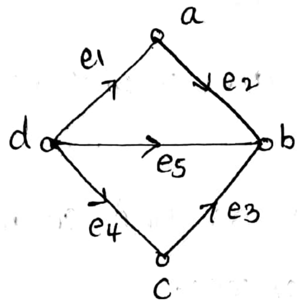
(2) Undirected graph

(1) Directed graph:

The graph in which the elements of The edge set are ordered pairs of vertices is called directed graph or digraph.

Here, order pair (v_i, v_j) denotes an edge from vertex v_i to vertex v_j. (v_j, v_i) denotes an edge from v_j to v_i.

eg:



directed graph

Here in this graph, elements edge set are ordered pair of vertices is,

$e_1 = (d, a)$ $\quad$ $e_4 = (d, c)$

$e_2 = (a, b)$ $\quad$ $e_5 = (d, b)$

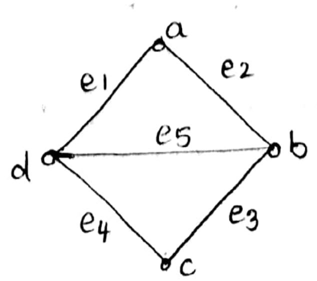
$e_3 = (c, b)$

(2) Undirected graph:

A graph in which the elements of the edge set are unordered pair of vertices is called undirected graph (non-directed).

Here (v_i, v_j) denotes an edge ~~from~~ between v_i, v_j.

eg:



undirected graph

Then $e_1 = \{a, d\}$ $e_4 = \{c, d\}$

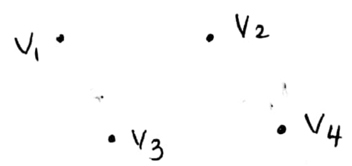
$e_2 = \{a, b\}$ $e_5 = \{b, d\}$

$e_3 = \{c, b\}$

* Null graph:

A graph in which number of edges is zero is called as Null graph.

eg:

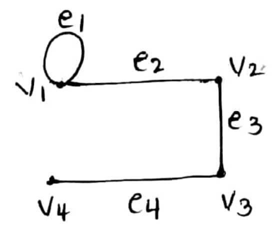


Null graph with 4 vertices and zero edges

* Selfloop:

An edge joining a vertex to itself is called as selfloop.

eg:



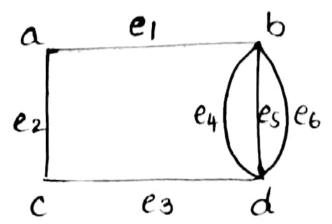
A graph with a selfloop

In this graph, edge e_1 is a self loop.

* Parallel or multiple edges:

In a graph it may be possible to have more than one edge with a single pair of vertices, such edges are called parallel edges.

eg:

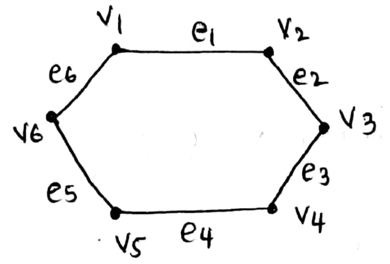


In this example e_4, e_5, e_6 are parallel edges.

* Simple graph:

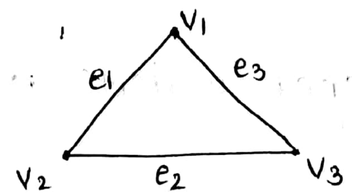
A graph which contains neither selfloop nor parallel edges is called a simple graph.

eg:



* Complete graph:

A simple graph in which there is exactly one edge between each pair of distinct vertices is called a complete graph. It is denoted by 'K_n'.



The no. of edges in completegraph with n vertices

$$= \frac{n(n-1)}{2}$$

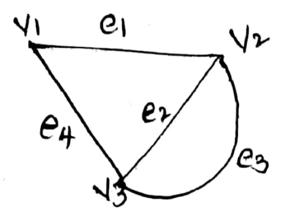
eg: Find the total number of edges of a complete graph with 50 vertices

The total number of edges of a complete graph with 50 vertices $= \frac{50(50-1)}{2} = 1225$.

* Multigraph:

A graph which contain parallel edges is called multigraph.

eg:



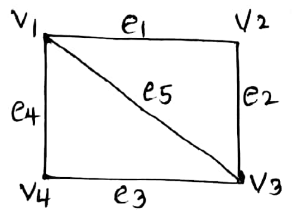
* order and size of a graph:

The number of vertices in a graph G is called order of the graph.

It is denoted by $|V(G)|$.

The number of edges in a graph G is called size of the graph and It is denoted by $|E(G)|$.

eg:



order of the graph is $|V(G)| = 4$

size of the graph is $|E(G)| = 5$

* Degree of vertex in a Non-directed graph and degree sequence:

The degree of a vertex v of a graph G is the number of edges of G, which are incident with v.

The degree of a vertex v in a graph G is denoted by $deg_G(v)$.

Isolated vertex:

A vertex of degree zero is called a isolated vertex.

Pendant vertex:

A vertex with degree one is called pendant vertex.

odd vertex:

A vertex with odd degree is called odd vertex.

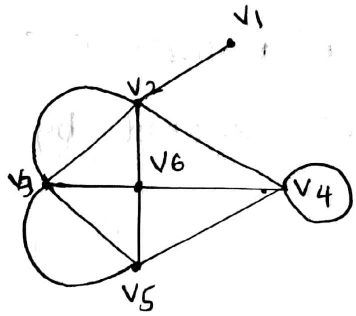
Even vertex:

A vertex of even degree is an even vertex.

NOTE: The degree of self loop is counted twice.

If $v_1, v_2, v_3 \ldots v_n$ are the vertices of G, then the sequence $\{d_1, d_2, d_3 \ldots d_n\}$ where $d_i = deg_G(v_i)$ is the degree sequence of G.

eg:



degree of vertex $v_1 = 1$

degree of vertex $v_2 = 5$

degree of vertex $v_3 = 5$

degree of vertex $v_4 = 5$

degree of vertex $v_5 = 4$

degree of vertex $v_6 = 4$

Degree sequence of graph is given by $\{1, 5, 5, 5, 4, 4\}$

* Degree of vertex in a directed graph

Indegree:
The number of edges incident to a vertex is called the Indegree of the vertex for a digraph.

Outdegree:
The number of edges incident from it is called the outdegree for a digraph.

The indegree of vertex v in a graph G is denoted by $deg_{G^+}(v)$.

The outdegree of vertex v in a graph G is denoted by $deg_{G^-}(v)$.

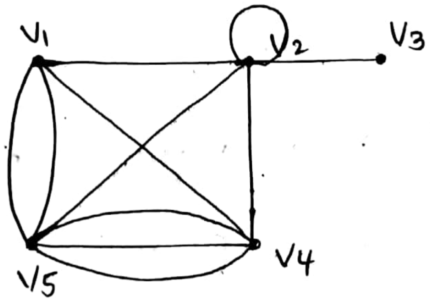
The degree of a vertex is determined by counting each loop incident on it twice and each other edge once.

The minimum of all the degrees of the vertices of a graph G is denoted by $\underline{\delta(G)}$.

The maximum of all the degrees of the vertices of a graph G is denoted by $\underline{\Delta(G)}$.

eg:

(1)

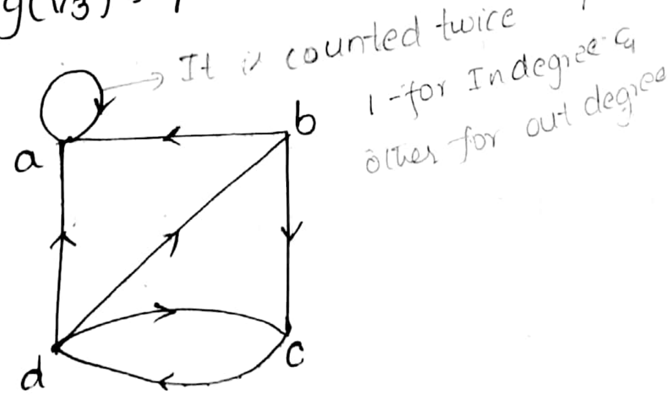


$deg(v_1) = 4$ $deg(v_4) = 5$

$deg(v_2) = 6$ $deg(v_5) = 6$

$deg(v_3) = 1$

(2)



$Deg^+(a) = 3$ $Deg^-(a) = 1$

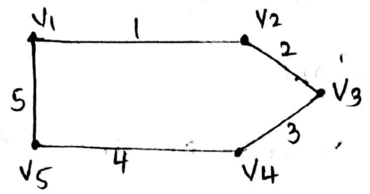
$Deg^+(b) = 1$ $Deg^-(b) = 2$

$Deg^+(c) = 2$ $Deg^-(c) = 1$

$Deg^+(d) = 1$ $Deg^-(d) = 3$

* **Weighted graph:**

A graph in which weights are assigned to every edge is called a weighted graph.



Here, 1, 2, 3, 4, 5 are weights assigned to each edge respectively.

* **Path:**

In a nondirected graph G, a sequence 'p' of zero or more edges of the form $\{v_0, v_1\}, \{v_1, v_2\} \ldots \{v_{n-1}, v_n\}$ or $v_0, v_1, v_2 \ldots v_n$ is called a path from v_0 to v_n. Where

v_0 is the initial vertex

v_n is the terminal vertex of the path p.

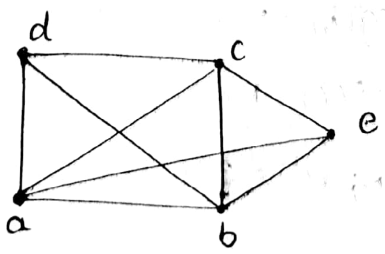
(1) In a path, vertices and edges may be repeated any number of times.

(2) The number of edges in a path is called length of the path.

Trivial path:

A path of length zero is called trivial path.

eg:



path	length
a-b-c-d	3
a-b-a	2
a-b-c-d-a	4
a-b-c-d-b-c-a	6
a	0
a-b	1

open path:

A path in which initial and terminal vertices are distinct is called open path.

eg: a-b-c-d is open path.

closed path:

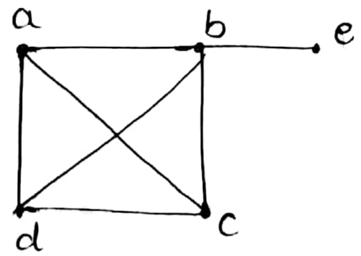
A path in which the initial and terminal vertices are same is called closed path.

eg: a, a-b-a, a-b-c-d-a, a-b-c-d-b-e-a are closed paths. (Trivial path is taken as closed path)

* simple path:

A path is said to be simple if all the edges and vertices on path are different except possibly of the end points.

eg:



simple path representation

Here the different paths are

(1) a-b-c-a

(2) a-b-c-d-c-a

(3) a-b-c-d-a

(4) a-b-c-a-b

Here (1) and (3) are simple paths, where as (2) and (4) are not simple paths

* Cycle graph:

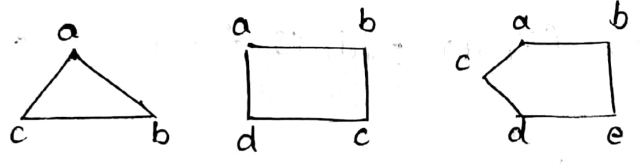
A cycle graph of order n is a connected graph whose edges forms a cycle of length 'n'.

It is denoted by 'C_n'.

(1) In a cycle graph of order 'n' vertices will have n vertices and n edges.

(2) In C_n, $deg(v_i) = n-1 \ \forall i$ and every C_n is a regular graph.

eg:



* Wheel graph:

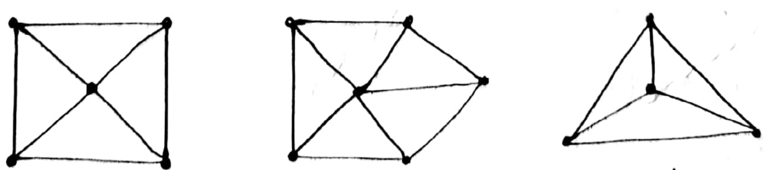
A wheel graph of order 'n' is a graph obtained by joining a single new vertex to each vertex of cycle graph (C_{n-1}) of order $(n-1)$.

It is denoted by W_n.

The number of vertices in a wheel graph is n+1 and edges 2n.

In $\boxed{W_n, \ |V| = n \quad |E| = 2(n-1)}$

eg:



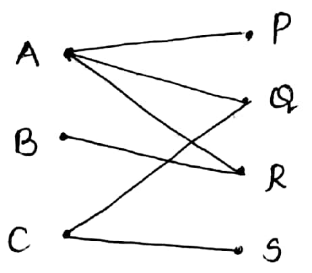
(a) W_5 b) W_6 (c) W_4

* V.imp Bipartite graph:

A simple graph G is such that its vertex set V is the union of two mutually disjoint non-empty sets V_1 and V_2 which are such that every edge in G joins a vertex in V_1 and a vertex in V_2. Then G is called a bipartite graph.

If E is the edge set of this graph, the graph is denoted by $G = (V_1, V_2; E)$ (or) $G = G(V_1, V_2; E)$.

The sets V_1 and V_2 are called bipartites (or partitions) of the vertex set V



In the above graph G,

vertex set $V = \{A, B, C, P, Q, R, S\}$

V is the union of V_1 and V_2

Edge set $E = \{AP, AQ, AR, BR, CQ, CS\}$

$V_1 = \{A, B, C\}$, $V_2 = \{P, Q, R, S\}$

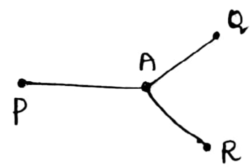
V_1 and V_2 are bipartites.

* Complete bipartite graph:

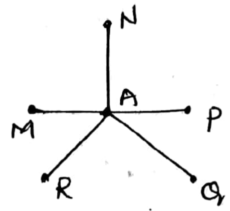
A bipartite graph $G = (V_1, V_2; E)$ is called a complete bipartite graph if there is an edge between every vertex in V_1 and every vertex in V_2.

A complete bipartite graph $G = (V_1, V_2; E)$ in which the bipartites V_1 and V_2 contains m and n vertices respectively with $m \leq n$ is denoted by $K_{m,n}$.

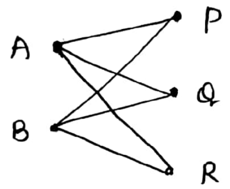
In this graph, each of m vertices in V_1 is joined to each of n vertices of V_2. Thus, $K_{m,n}$ has m+n vertices and mn edges i.e; $K_{m,n}$ is of order m+n and size mn; It is therefore a (m+n, mn) graph.



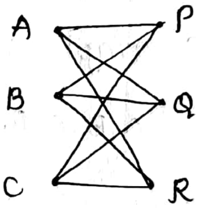
(a) $K_{1,3}$



(b) $K_{1,5}$



(c) $K_{2,3}$



(d) $K_{3,3}$

(a) → The bipartities are $V_1 = \{A\}$ and $V_2 = \{P, Q, R\}$
The vertex A is joined to each of the vertices P, Q, R by an edge.

(b) The bipartities are $V_1 = \{A\}$ and $V_2 = \{M, N, P, Q, R\}$
The vertex A is joined to each of the vertices M, N, P, Q, R by an edge.

(c) The bipartities are $V_1 = \{A, B\}$ and $V_2 = \{P, Q, R\}$ each of the vertices A and B are joined to each of the vertices P, Q, R by an edge.

(d) The bipartites are $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R\}$
Each of the vertices A, B and C is joined to each of the vertices P, Q, R.

The graph $K_{3,3}$ is of great important. This is known as the Kuratowski's second graph

* Representation of Graphs:

A matrix is a convenient and useful way of representing a graph. Many known results of matrix algebra can be applied to study the properties of graphs and to calculate paths, cycles and other characteristics of a graph.

We have 3 types for representing the graphs

(1) Adjacency matrix
(2) Incidence matrix
(3) pathmatrix of a graph.

(1) Adjacency matrix : (undirected graph)

Let $G(V, E)$ be a simple graph with n vertices ordered from v_1 to v_n, then the adjacency matrix

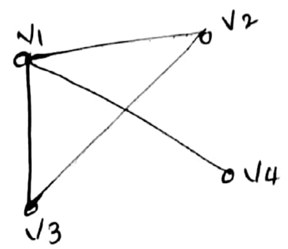
$$A = [a_{ij}]_{n \times n}$$

of G is an $n \times n$ symmetric matrix defined by the elements

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

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

eg: A graph G and its adjacency matrix A_G are shown below.



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

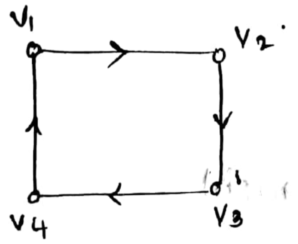
Adjacency matrix in case of directed graph:

Let $G(V, E)$ be a simple digraph with $V = \{v_1, v_2 -- v_n\}$ and the vertices are assumed to be ordered from v_1 to v_n. An $n \times n$ matrix A whose elements a_{ij} are given by

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

is called adjacency matrix of the graph G.

eg:

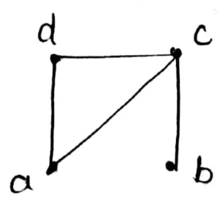


$$A_G = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Problems:

(1) Draw a graph with the given adjacency matrix

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



(2) Without drawing the graph, prove that the graph whose adjacency matrix is given by

$$X = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$ is connected.

Here, the adjacency matrix is given by

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

This is of the order $n=5$

We compute x^2, x^3, x^4

$$X^2 = \begin{bmatrix} 2 & 1 & 0 & 1 & 2 \\ 1 & 3 & 1 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 3 & 1 \\ 2 & 1 & 0 & 1 & 3 \end{bmatrix} \qquad X^3 = \begin{bmatrix} 2 & 5 & 2 & 5 & 2 \\ 5 & 4 & 1 & 5 & 6 \\ 2 & 1 & 0 & 1 & 3 \\ 5 & 5 & 1 & 4 & 6 \\ 2 & 6 & 3 & 6 & 2 \end{bmatrix}$$

$$X^4 = \begin{bmatrix} 10 & 9 & 2 & 9 & 12 \\ 9 & 16 & 6 & 15 & 10 \\ 2 & 6 & 3 & 6 & 2 \\ 9 & 15 & 6 & 16 & 10 \\ 12 & 10 & 2 & 10 & 15 \end{bmatrix}$$

Every entry in x^4 is non-zero. Therefore, no entry in $y = x + x^2 + x^3 + x^4$ can be zero.

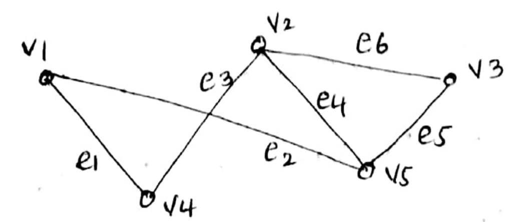
Hence, the given graph is connected.

(2) Incidence matrix (undirected graph)

Let G be a graph with n vertices. Let $V = \{v_1, \dots v_n\}$ and $E = \{e_1, e_2, \dots e_m\}$. Define $n \times m$ matrix $I_G = [m_{ij}]_{n \times m}$

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

eg:



Scanned with CamScanner

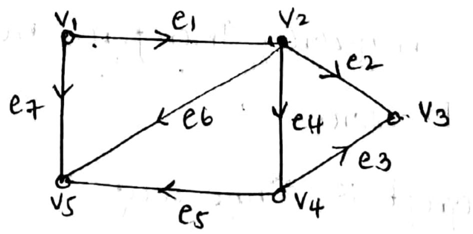
$$I_G = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

In case of directed graph

In Incidence matrix with the vertex set $\{v_1, v_2 \ldots v_n\}$ the edgeset $\{e_1, e_2 \ldots \cancel{e_m}\}$ and with no self loops is an $n \times m$ matrix $B = (b_{ij})$ defined by

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

eg: A digraph G and its incidence matrix B_G are given as follows :



incoming = -1
outgoing = 1

$$B(G) = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} +1 & 0 & 0 & 0 & 0 & 0 & +1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix} \end{array}$$

(3) Pathmatrix / reachability matrix.

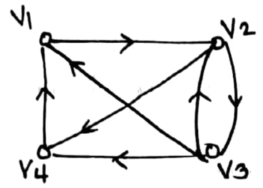
Let G be a simple digraph having no parallel directed edges and $V = \{v_1, v_2, \ldots v_n\}$ be its vertex set. An $\overset{n}{\cancel{m}} \times n$ matrix $P = [P_{ij}]_{\cancel{m}n \times n}$ is given by

let $G = \langle V, E \rangle$ be a simple digraph in which $|V| = n$ and the nodes of G are assumed to be ordered. An $n \times n$ matrix P whose elements are given by

$$P_{ij} = \begin{cases} 1 & \text{if there is a path from } v_i \text{ to } v_j \\ 0 & \text{otherwise} \end{cases}$$

is called a path matrix of G

eg: Consider the graph find path matrix



Adjacency matrix of the graph

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

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad A^4 = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 2 & 2 & 3 \\ 3 & 3 & 2 & 3 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

$B_n = A + A^2 + A^3 + \dots + A^n$

Now $B = A + A^2 + A^3 + A^4$

Then $B = \begin{bmatrix} 3 & 4 & 2 & 3 \\ 5 & 5 & 4 & 6 \\ 7 & 7 & 4 & 7 \\ 3 & 2 & 1 & 2 \end{bmatrix}$

All vertices in the graph are connect with different paths.

$$\therefore \quad P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

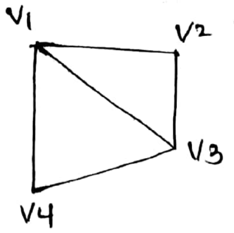
* Subgraphs:

Given two graphs G and G_1, we say that G_1 is a subgraph of G. If it follows conditions that,

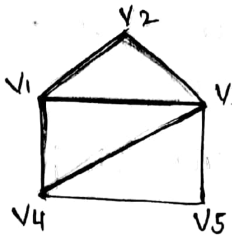
(1) All the vertices and all the edges of G_1 are in G.

(2) Each edge of G_1 has the same end vertices in G as in G_1.

eg:



(a): G_1



(b): G

From the two graphs, we observe that all vertices and all edges of the graph G_1 are in graph G and that every edge in G_1 has the same end vertices in G as in G_1.

Therefore, G_1 is a subgraph of G.

Note:

(1) Every graph is a subgraph of itself

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

(3) If G_1 is a subgraph of graph G_2 and G_2 is a subgrap of a graph G, then G_1 is a subgraph of G.

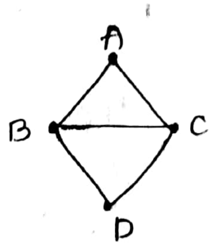
(4) A single vertex in a graph G is a subgraph of G.

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

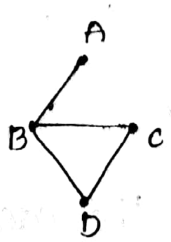
* Spanning subgraph:

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

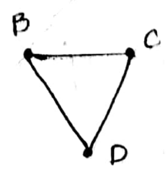
eg:



(a)



(b)



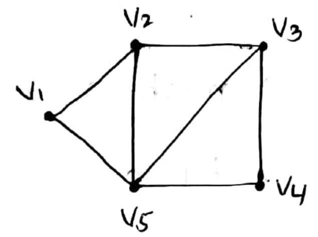
(c)

from the above, we observe that, fig(a) is the graph, fig(b) is a spanning subgraph whereas the graph (c) is a subgraph but not a spanning subgraph. Because fig(b) contains all vertices of (a) but, (c) does not contains all the vertices of (a).

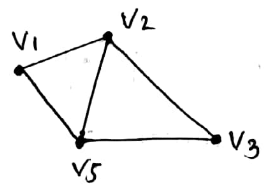
* Induced subgraph :

Given a graph $G = (V, E)$, suppose there is a subgraph $G_1 = (V_1, E_1)$ of G such that every edge $\{A, B\}$ of G Where $A, B \in V_1$ is an edge of G_1 also.

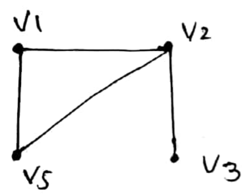
Then, G_1 is called a subgraph of G induced by V_1 and denoted by $\langle V_1 \rangle$.



(a)



(b)



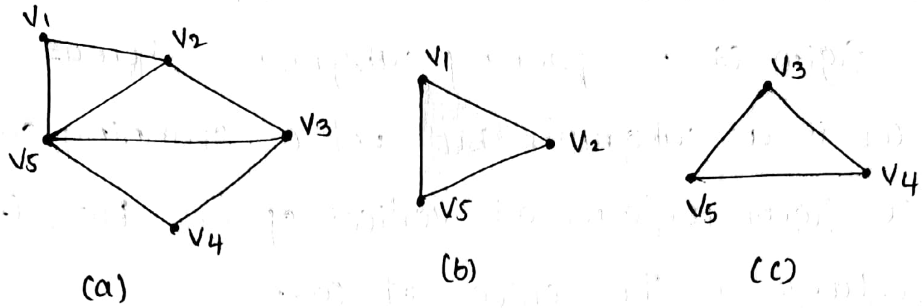
For the graph (a), the graph (b) is an induced subgraph - induced by the set of vertices $V_1 = \{v_1, v_2, v_3, v_5\}$ whereas the graph (c) is not an induced subgraph.

* Edge-disjoint and vertex-disjoint subgraphs :

Let G be a graph and G_1 and G_2 be two subgraphs of G. Then:

(1) G_1 and G_2 are said to be "edge-disjoint" if they donot have any common edge.

(2) G_1 and G_2 are said to be "vertex-disjoint" if they donot have any common edge and any common vertex.



(a) (b) (c)

For the graph (a), the graphs (b) and (c) are edge-disjoint but not vertex-disjoint subgraphs.

* Complement of a graph:

Every simple graph of order n is a subgraph of the complete graph K_n. If G is a simple graph of order n, Then the complement of G in K_n is called the "complement of G". It is denoted by $\bar{G}$.

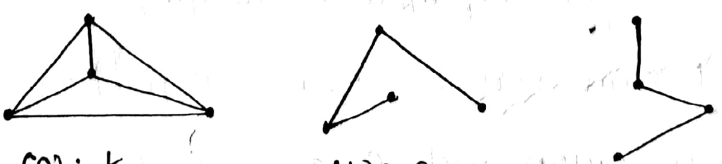
Thus, The complement $\bar{G}$ of a simple graph G with n vertices is that graph which is obtained by deleting those edges of K_n, which belongs to G.

Thus,

$$\bar{G} = K_n - G$$

$$\bar{G} = K_n \Delta G$$

eg:



(a) : K_n (b): G (c) : $\bar{G}$

from the above, The graph (a) is the complete graph K_4. A simple graph G of order 4 is in graph (b). The complement $\bar{G}$ of G is shown in (c)

Observe that G, $\bar{G}$ and K_4 have the same vertices and that the edges of $\bar{G}$ are got by deleting those edges from K_4 which belongs to G.

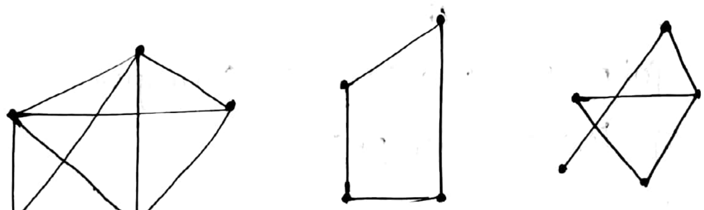
Scanned with CamScanner

* Complement of a subgraph:

Given a graph G and a subgraph G_1 of G, the subgraph of G obtained by deleting from G all the edges that belong to G_1 is called "complement of G_1 in G".
It is denoted by $G-G_1$ (or) $\overline{G_1}$.

In otherwords, if E_1 is the set of all edges of G_1, Then the complement of G_1 in G is given by

$\overline{G_1} = G - E_1$.



(a) (b) (c)

For example, consider the graph G shown in (a). G_1 be the subgraph of G shown in (b). The complement of G_1 in G; namely $\overline{G_1}$ is shown in graph(c).

VVVImp

* Isomorphism:

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs.
A function $f: G_1 \to G_2$ is called an isomorphism. If
(1) If f is one to one and onto
(2) If the graph G_1 is isomorphic to G_2 then we write

$G_1 \cong G_2$.

properties:

If 2 graphs G_1 and G_2 are isomorphic then

(a) $|V(G_1)| = |V(G_2)|$

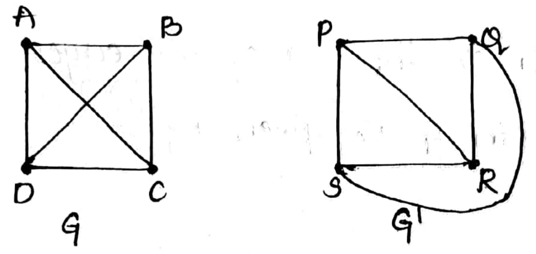
(b) $|E(G_1)| = |E(G_2)|$

(c) $\deg_{G_1}(v) = \deg_{G_2}(v)$.

i.e; The degree sequences of G_1 and G_2 are the same.

(d) If 2 graphs are isomorphic then their adjacency matrices are same.

for example, let us consider the graphs:



(a) (b)

consider the following one-to-one correspondence between the vertices of these 2 graphs

$A \leftrightarrow P$

$B \leftrightarrow Q$

$C \leftrightarrow R$

$D \leftrightarrow S$

under this correspondence, the edges in the two graphs corresponds with each other as indicated below.

$\{A,B\} \leftrightarrow \{P,Q\}$ $\{B,C\} \leftrightarrow \{Q,R\}$

$\{A,C\} \leftrightarrow \{P,R\}$ $\{B,D\} \leftrightarrow \{Q,S\}$

$\{A,D\} \leftrightarrow \{P,S\}$ $\{C,D\} \leftrightarrow \{R,S\}$

The number of vertex in the 2 graphs are same.

i.e; $V(G) = 4$

$V(G') = 4$

$\therefore \boxed{V(G) = V(G')}$

The number of edges in the 2 graphs are same

i.e; $E(G) = 6$ $E(G') = 6$

$\therefore \boxed{E(G) = E(G')}$

The degree sequences of G and G' are same

$\deg G(v) = (3,3,3,3)$

$\deg G'(v) = (3,3,3,3)$

In the 2 graphs, adjacency matrices are equal or same.

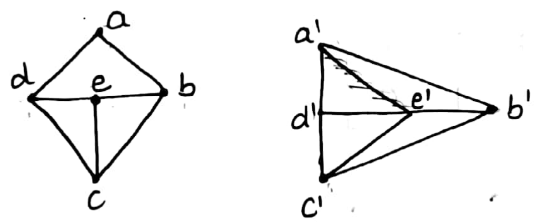
$$G = \begin{array}{c c} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$G' = \begin{array}{c c} & \begin{array}{cccc} P & Q & R & S \end{array} \\ \begin{array}{c} P \\ Q \\ R \\ S \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$\therefore$ The given 2 graphs are Isomorphic.

* Problems:

(1) Determine whether the following graphs are isomorphic



G_1 G_2

Consider the following one-to-one correspondence b/w the vertices of these 2 graphs.

$a \leftrightarrow a'$

$b \leftrightarrow b'$

$c \leftrightarrow c'$

$d \leftrightarrow d'$

under this correspondence, the edges in the 2 graphs corresponds with each other as indicated below

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

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

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

The number of vertex in 2 graphs are same

ie; $V(G_1) = V(G_2)$

$V(G_1) = 5$

$V(G_2) = 5$

The number of edges in 2 graphs are same

i.e; $E(G_1) = 7$

$E(G_2) = 7$

$\therefore E(G_1) = E(G_2)$

The degree sequences of G_1 and G_2 are same

$\deg G_1(V) = (2, 3, 3, 3, 3)$

$\deg G_2(V) = (2, 3, 3, 3, 3)$

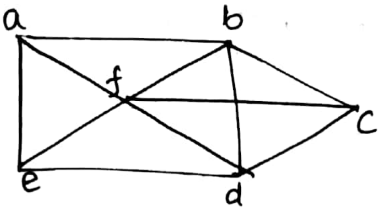
In the 2 graphs adjacency matrices are equal or same

$$G_1 = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 0 & 1 & 0 \\ b & 1 & 0 & 1 & 0 & 1 \\ c & 0 & 1 & 0 & 1 & 1 \\ d & 1 & 0 & 1 & 0 & 1 \\ e & 0 & 1 & 1 & 1 & 0 \end{array}$$

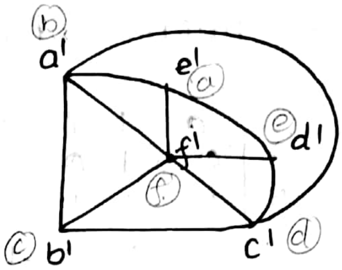
$$G_2 = \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 0 & 1 & 0 \\ b & 1 & 0 & 1 & 0 & 1 \\ c & 0 & 1 & 0 & 1 & 1 \\ d & 1 & 0 & 1 & 0 & 1 \\ e & 0 & 1 & 1 & 1 & 0 \end{array}$$

$\therefore$ The given 2 graphs are isomorphic.

(2)



(a) Graph (G)



(b) graph (G')

Consider the following one-to-one correspondence b/w the vertices of 2 graph

$a \leftrightarrow e'$ $\quad e \leftrightarrow d'$

$b \leftrightarrow a'$ $\quad f \leftrightarrow f'$

$c \leftrightarrow b'$

$d \leftrightarrow c'$

under this correspondence, the edges in the 2 graphs corresponds with each other as indicated below

$\{a,b\} \leftrightarrow \{e',a'\}$ $\quad \{c,f\} \leftrightarrow \{b',f'\}$

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

$\{a,f\} \leftrightarrow \{e',f'\}$ $\quad \{d,e\} \leftrightarrow \{c',d'\}$

$\{b,f\} \leftrightarrow \{a',f'\}$ $\quad \{d,f\} \leftrightarrow \{c',f'\}$

$\{b,d\} \leftrightarrow \{a',c'\}$ $\quad \{e,f\} \leftrightarrow \{d',f'\}$

$\{b,c\} \leftrightarrow \{a',b'\}$

The number of vertex in 2 graphs are same

i.e; $V(G) = 6$

$V(G') = 6$

$\therefore V(G) = V(G')$

The number of edges in 2 graphs are same

i.e; $E(G) = 11$

$E(G') = 11$

The degree sequences of G and G' are same

$\deg G(v) = (3,4,3,4,3,5)$

$\deg G'(v) = (3,4,3,4,3,5)$

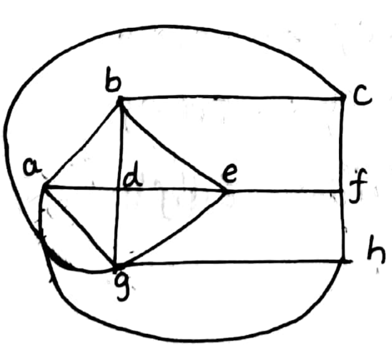
In the 2 graphs adjacency matrices are equal or same

$$G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

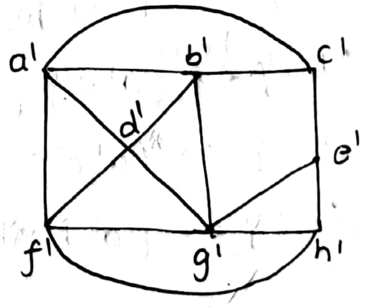
$$G' = \begin{array}{c} \\ a' \\ b' \\ c' \\ d' \\ e' \\ f' \end{array} \begin{array}{c} \begin{array}{cccccc} a' & b' & c' & d' & e' & f' \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$\therefore$ The given 2 graphs are isomorphic

(3)

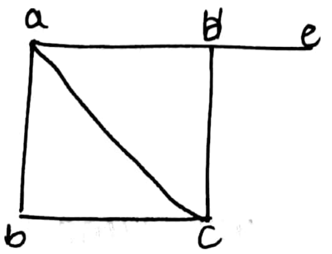


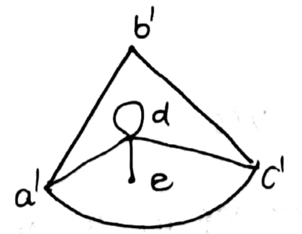
a) Graph (G)



(b) Graph (G')

4)





a) graph (G)

* Walks and their classification : (vertex Degree)

There are 5 important subgraphs of a graph

(1) Walk
(2) Trail
(3) Circuit
(4) path
(5) cycle

1) Walk:

Consider a graph and having atleast one edge. In G, alternating sequence of vertices and edges of the form

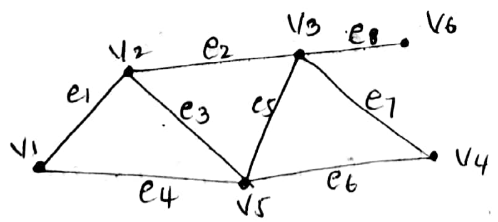
$$V_i\, e_j\, V_{i+1}\, e_{j+1}\, V_{i+2}\, e_{j+2} \cdots e_k V_m$$

Which begins and ends with vertices and which is such that each edge in the sequence is incident on the vertices. such sequence is called a walk.

In a walk, a vertex or an edge (or both) can appear more than once.

The number of edges present in a walk is called its length.

eg:



In this graph

(1) The sequence $V_1 e_1 V_2 e_2 V_3 e_8 V_6$ is a walk of length 3 (because, this walk contains 3 edges : e_1, e_2, e_8). In this walk no vertex and no edge is repeated.

(2) The sequence $V_1 e_4 V_5 e_3 V_2 e_2 V_3 e_5 V_5 e_6 V_4$ is a walk of length 5. In this walk, the vertex V_5 is repeated, but no edge is repeated.

(3) The sequence $V_1 e_1 V_2 e_3 V_5 e_3 V_2 e_2 V_3$ is a walk of length 4. In this walk, the edge e_3 is repeated and the vertex V_2 is repeated.

The vertex with which a walk begins is called the "initial vertex" (or the origin) of the walk and the vertex with which a walk ends is called the "final vertex" of the walk.

Closed walk:

A walk that begins and ends at the same vertex is called a closed walk.

Open walk:

A walk that is not closed is called an open walk.

eg: $V_1 e_1 V_2 e_3 V_5 e_4 V_1$ - closed walk.

$V_1 e_1 V_2 e_2 V_3 e_5 V_5$ - open walk.

(2) Trail and circuit:

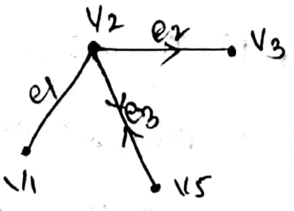
Trail:

In a walk, vertices and edges may appear more than once. If in an open walk no edge appears more than once, then walk is called a trail.

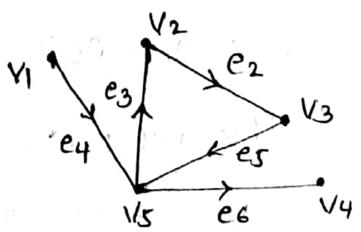
Circuit:

A closed walk in which no edge appears more than once is called a circuit.

eg:

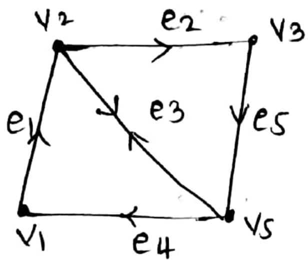


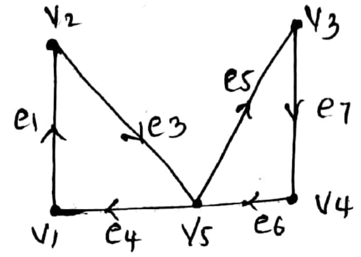
(a): Not a trail



(b): Trail

From the graph (a) is not a trail i.e; the open walk $v_1 e_1 v_2 e_3 v_5 e_3 v_2 e_2 v_3$, because e_3 is repeated. where and graph (b) is trail.





From the above, the closed walk $v_1 e_1 v_2 e_3 v_5 e_5 v_3 e_7 v_4 e_6 v_5 e_4 v_1$ is a circuit

Path and cycle:

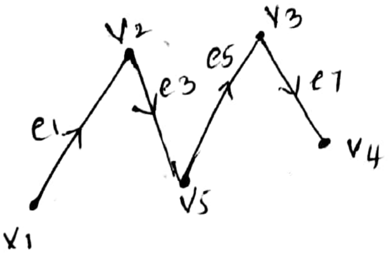
Path:

A Trail in which no vertex appears more than once is called a path.

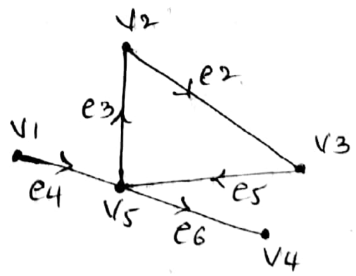
Cycle:

A circuit in which the no vertex appears more than once except end vertices. is called a cycle.

eg:

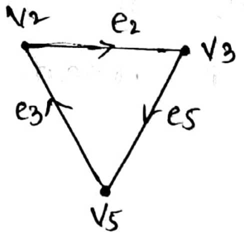


(a) path

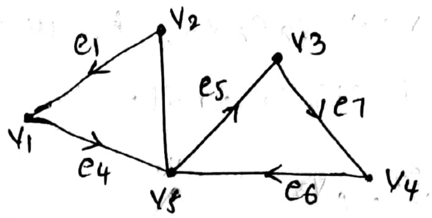


(b) Not a path

From the above, the trail $v_1 e_1 v_2 e_3 v_5 e_5 v_3 e_7 v_4$ i.e; graph (a) is a path where as the trail $v_1 e_4 v_5 e_3 v_2 e_2 v_3 e_5 v_5 e_6 v_4$ (i.e; (b)) is not a path (because in this trail, v_5 appears twice).



(c): cycle



(d): Not a cycle

The circuit $V_2 e_2 V_3 e_5 V_5 e_3 V_2$ (graph(c)) is a cycle where as the circuit $V_2 e_1 V_1 e_4 V_5 e_5 V_3 e_7 V_4 e_6 V_5 e_3 V_2$ (graph d) is not a cycle. (because in this circuit V_5 appears twice).

VVImp

* **Euler's circuits and Euler's trails**

Consider a connected graph G. If there is a circuit in G that contains all the edges of G then that circuit is called "Euler's circuit" (or Eulerian line or Eulertour)

Euler trail:

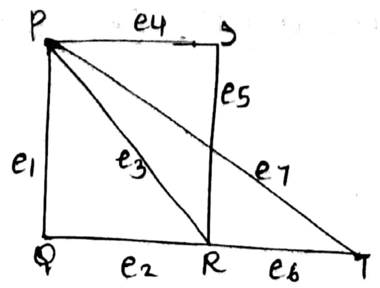
If there is a trail in G that contains all the edges of G, then that trail is called an Euler's trail (or Unicuisal line) in G.

A connected graph that contains an "Euler circuit" is called Eulergraph (or) Eulerian graph.

Semi- Euler's graph:

A connected graph that contains an "eulertrail" is called a Semi- Eulor's graph.

eg:

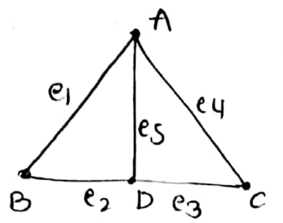


(a)

In the graph (a), the closed walk

$P e_1 Q e_2 R e_3 P e_4 S e_5 R e_6 T e_7 P$

is a Euler circuit



The trail $A e_1 B e_2 D e_3 C e_4 A e_5 D$ in the graph is an Euler trail. This graph is a semi-euler graph.

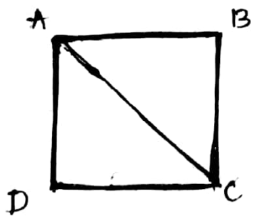
* Hamilton cycles and Hamilton paths.

Let G be a connected graph. If there is a cycle in G that contains all the vertices of G, then that cycle is called a "Hamilton cycle" in G.

A Hamilton cycle in a graph of n vertices consists of exactly n edges. Because, a cycle with n vertices has n edges.

A graph that contains a hamilton cycle is called a "Hamilton graph" (or Hamitonian graph)

eg:



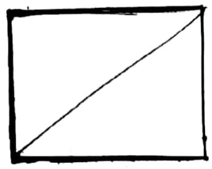
In the above graph, the cycle shown in thick lines is a Hamilton cycle (observe that this cycle doesnot include the edge BD). The graph is therefore a hamilton.

Let G = (V, E) be an undirected graph or multigraph with no isolated vertices. Then G is said to be euler circuit if there is a circuit in G that traverses every edge of the graph exactly once. If there is an open trail from a to b in G & this trail traverse each edge in G exactly once, that trail is called euler trail

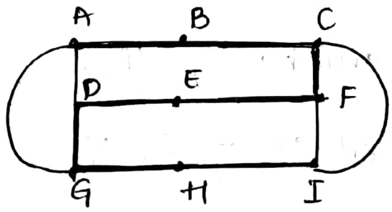
Hamiltonpath:

A path in a connected graph which includes every vertex (but not necessarily every edge) of the graph is called a "hamilton/ hamilton path in, The graph.

eg:



In the above graph, the path shown in thick line is hamilton path.



In above graph, the path ABCFEDGHI is a hamiltonpath. We check that this graph does not contain a hamilton cycle.

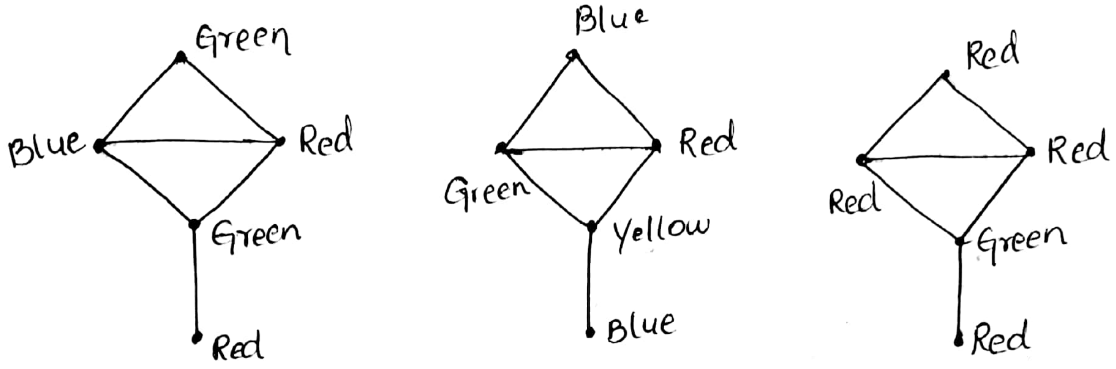
Note:

A path with n vertices has n-1 edges.

* Graph coloring:

Given a planar or non-planar graph G, if we assign colors (colours) to its vertices in such a way that no two adjacent vertices have the same colour, then we say that the graph G is properly coloured.

In otherwords, proper coloring of a graph means assigning colours to its vertices such that adjacent vertices have different colours.



From the above graph, the first 2 graphs are properly coloured whereas the 3rd graph is not properly coloured.

By examining the ~~firs las~~ first two graphs, which are properly colored, we note the following.

(1) A graph can have more than one proper coloring.

(2) Two non-adjacent vertices in a properly colored graph can have the same colour.

* Chromatic number:

A K-chromatic graph that can be properly coloured with K colors but not with less than K colours.

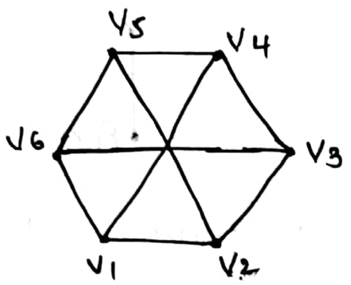
If a graph G is K-chromatic, then K is called the "chromatic number" of G.

Thus, the chromatic number of graph is the minimum number of colors with which the graph can be properly colored.

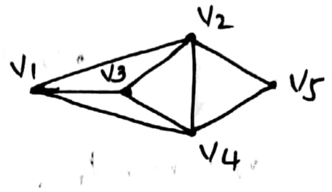
The chromatic number of a graph G is usually denoted by $\chi(G)$.

eg:

1) Find the chromatic number of each of the following graphs.



(a)



(b)

(i) For the graph (a), let us assign a color α to the vertex v_1 then, for a proper colouring, we have to assign a differcolour to its neighbours v_2, v_4, v_6, Since v_2, v_4, v_6 are mutually non adjacent vertices, they have the same colour, say β (which is different from α). Since v_3, v_5 are not adjacent to v_1, these can have the same colour as v_1, namely α.

Thus, the graph can be properly colored with atleast two colours. With the vertices v_1, v_3, v_5 having one colour α and v_2, v_4, v_6 having a different color β. Hence, the chromatic number of the graph is 2.

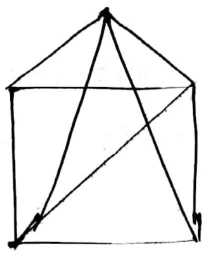
(ii) For the graph (b), let us assign the colour α to the vertex v_1 then, for a proper colouring, its neighbours v_2, v_3 and v_4 cannot have the colour α, but v_5 can have the colour α. Furthermore, v_2, v_3, v_4 must have different colours, say β, γ, δ. Thus, atleast four colours are required for a proper colouring of the graph. Hence, the chromatic number of the graph is 4.

* Planar Graphs :

A graph G is called planar, if it can be drawn on the plane in such a way that no two edges cross each other at any point ; except possibly at the common end vertex. such a drawing is called plane drawing.

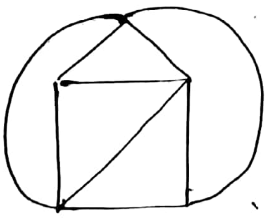
eg:

(1) Consider the graph as shown in fig

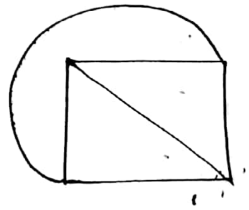


Graph (G)

It's planar graphs are,

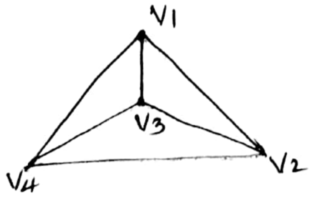


(a) planar graph (G₁)

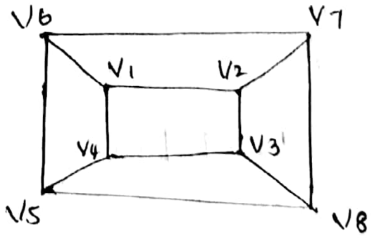


(b) planar graph (G')

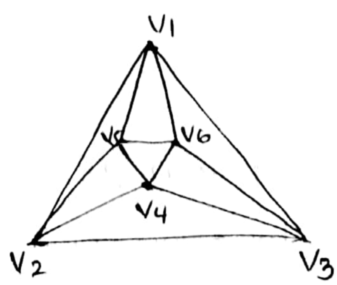
(2) Graphs of regular solids



(a) Tetrahedron



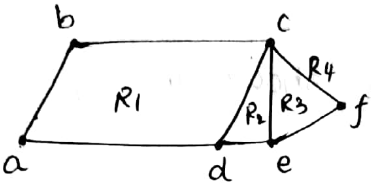
(b) Cube



(c) Octahedron

* Region (or) Face of a graph:

If a connected, planar graph is drawn in the same plane, the plane is divided into the continguous regions called faces. A face is characterized by the cycle that forms its boundary.



Face R_1 is bounded by the cycle (a, b, c, d, a)

Face R_2 is bounded by the cycle (c, d, e, c)

Face R_3 is bounded by the cycle (c, e, f, c).

The outer region R_4 is bounded by the cycle (a, b, c, f, e, d, a).

$f = 4$ faces, $e = 8$ edges, $v = 6$ vertices

→ $\deg(R_1) = 4$

→ $\deg(R_2) = 3$

→ $\deg(R_3) = 3$

→ $\deg(R_4) = 6$

TB 19:562/563

* Euler's formula:

Theorem:

(1) If G is a connected plane graph, then

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

(or)

for any planar graph G, number of vertices - number of edges + number of regions = 2.

Proof:

Given a connected planar graph G, then any drawing of G in the plane, as plane graph will always from $|R| = |E| - |V| + 2$ regions, including exterior regions.

Here,

$|R|$ = number of regions

$|E|$ = Number of edges

$|V|$ = Number of vertices.

This formula was discovered by Euler, hence their name Euler's formula. Now we will discuss this in detail

Statement :

If G is a connected plane graph, then $|V|-|E|+|R| = 2$

Proof:

We prove it by first observing the result for a tree. A tree determines only 1 region. If the number of vertices in a tree are n, the edges will be n-1. Therefore the formula $|V|-|E|+|R| = 2$ holds good for tree

i.e; $|V|-|E|+|R| = n-(n-1)+1$

$= \cancel{n}-\cancel{n}+1+1$

$= 2$

Therefore, the results is true for $r=1$, suppose that the result is true for k regions and suppose that G is a connected plane graph that has K+1 regions.

Delete one edge common to the boundary of two separate regions.

Then the resulting graph G_1 has the same number of vertices, one fewer edge and one fewer region because the deletion of an edge makes 2 regions into one region.

Thus, if $|E_1|$, $|V_1|$ and $|R_1|$ are respectively the number of edges, vertices and regions for G_1 then

$|E_1| = |E| - 1$

$|V_1| = |V|$

$|R_1| = |R| - 1$

$\therefore \; |V_1| - |E_1| + |R_1| = |V| - (|E| - 1) + (|R| - 1)$

$= |V| - |E| + \cancel{1} + |R| - \cancel{1}$

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

$\therefore \; |V_1| - |E_1| + |R_1| = |V| - |E| + |R|$

By inductive hypothesis

$|V_1| - |E_1| + |R_1| = 2$

$\therefore \; |V| - |E| + |R| = 2$

Thus, the theorem is proved by mathematical induction.

* Dual of a planar Graph:

Consider a connected planar graph G and a plane drawing. suppose R_1, R_2, R_3, etc be the regions (including the exterior region) in this drawing. Let us now construct a graph G^* following the procedure given below.

(1) Choose one point inside each of the regions $R_1, R_2, R_3, \ldots$ denote these point by $v_1^*, v_2^*, v_3^*, \ldots$ These parts will be the vertices of G^*.

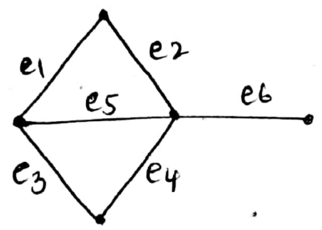
(2) If two regions R_i and R_j are adjacent (ie; have a common edge, say e_k), draw a line e_k^* joining the points v_i^* and v_j^* that intersects the common edge e_k exactly once.

(3) If there is more than one edge common to R_i and R_j, draw one line e_p^* b/w the points v_i^* and v_j^* for each common edge e_p, intersecting e_p exactly once.

4) For an edge e_i lying entirely in one region, say R_i draw a loop e_i^* at the point v_i^* intersecting e_i exactly once.

The graph G^* so constructed is called a "geometric dual" or just a "dual" of G.

eg: Consider the graph G shown in fig



We observe that, the fig. divides the plane into 3 regions R_1, R_2 and R_3 of which R_3 is unbounded.

We construct the dual G^* of G.

The step-by-step description of the construction is given below.

→ We choose 3 points v_1^*, v_2^*, v_3^* inside the regions R_1, R_2, R_3 respectively.

→ The regions R_1 and R_2 have a common edge e_5 we draw a line e_5^* joining v_1^* and v_2^* that crosses e_5 exactly once.

→ The regions R_1 and R_3 have 2 common edges e_1 and e_2. We draw 2 lines e_1^*, e_2^* b/w v_1^* and v_3^* with e_1^* crossing only e_1 and e_2^* crossing only e_2.

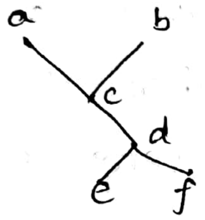
Trees :

Definition :

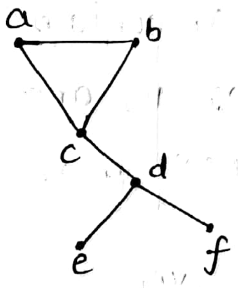
Let $G=(V,E)$ has be a loop-free, undirected graph. The graph G is called a tree if G is connected and contains no cycles.

eg:

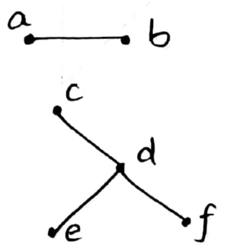
The graph G_1 is tree, but the graph G_2 is not a tree because it contains the $\{a,b\}, \{b,c\}, \{c,a\}$



G_1



G_2



G_3

The graph G_3 is not connected, so it cannot be a tree

Spanning trees :

A subgraph H of a graph G is called a spanning-tree If

(a) H is a tree, and

(b) H contains all the vertices of G.

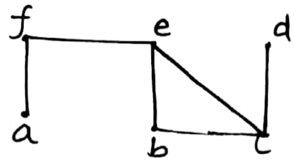
A spanning tree that is a directed tree is called a directed spanning tree of G.

Spanning tree can be generated by both the traversals of a graph i.e Depth first search (DFS) and Breadth first search.

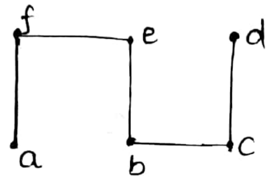
Scanned with CamScanner

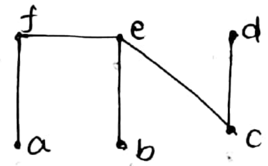
eg:

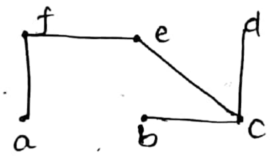
1. find all spanning trees for the graph G and shown in fig



The graph G is connected. It has 6 edges and 6 vertices and hence each spanning tree have $n-1$ edges ie; $6-1=5$ edges.







Note: In general, if G is a connected graph with n vertices and m edges, a spanning tree of G must have $n-1$ edges

Properties :

(1) If a, b are distinct vertices in a tree $T=(V,e)$, then there is a unique path that connects these vertices.

(2) If $G=(V,E)$ is a undirected graph, then G is connected if and only if G has a spanning tree.

(3) In every tree $T=(V,E)$, $|V| = |E|+1$

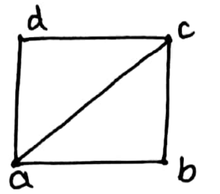
ie; $V=n$, $E=n-1$

(4) For every tree $T=(V,E)$, if $|V| \geq 2$, then T has atleast two pendant vertices.

(5) $m-n+1$ edges must be removed from a connected graph with n vertices and m edges to produce a spanning tree.

example :

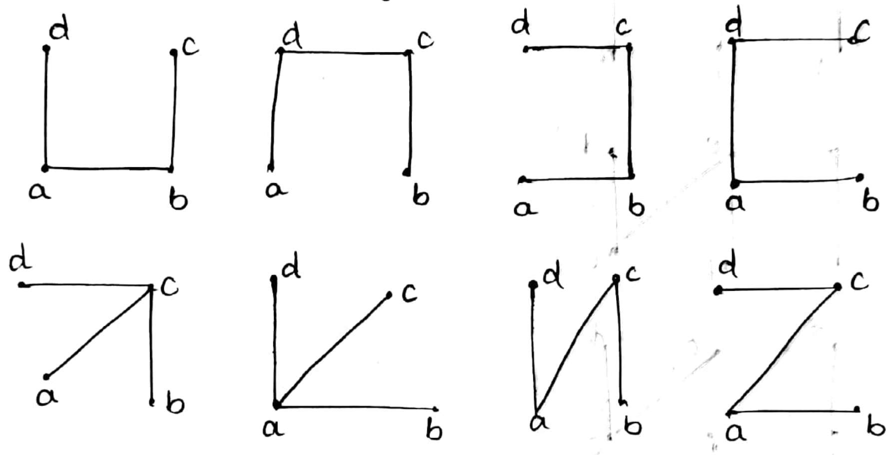
(1) Draw all the spanning trees of the tree graph.



This graph has $m = 5$ edges, $n = 4$ vertices

To obtain a spanning tree we should remove

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



Graph traversals :

Given a graph $G = (V, E)$ and a vertex V in G we are intersetted in visiting all vertices in G that are reachable from (V) (ie; all vertices connected to V).

There are 2 ways of doing this

(1) Depth first search (Back tracking)

(2) Breadth first search.

(1) Depth first search :

General procedure :

Let $G = (V, E)$ be a connected graph of order n, with vertices labelled $v_1, v_2, \ldots, v_n$ in some specified order. The DFS algorithm specifies the following steps for determining a spanning tree T of G where in the varcable v stands for the vertex being considered.

Step 1:

Assign the first vertex v_1 to the variable v and initialize T at the tree consisting of just this vertex.

Step 2:

Select the smallest subscript k for $2 \le k \le n$, such that $\{v, v_k\} \in E$ and v_k has not already been included in T.

If no such subscript is found, then go to step 3 otherwise perform the following

(i) Attach the edge $\{v, v_k\}$ to T

(ii) Assign v_k to v.

(iii) Return to step 2.

Step 3:

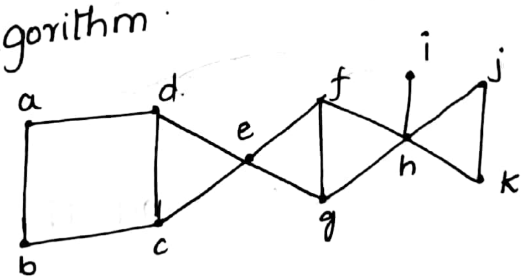
If $v = v_1$, the tree T is the spanning tree for the order specified.

Step 4:

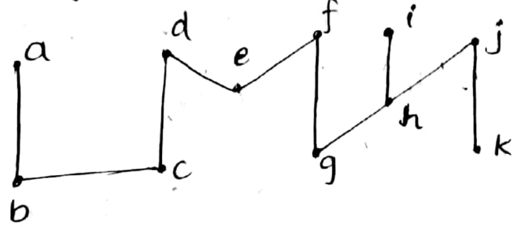
for $v \neq v_1$, backtrack from v. If u is the parent of the vertex assigned to v to T, then assign u to v and return to step 2.

example:

(1) Obtain a spanning tree of a following graph using DFS algorithm. step by step



The DFS spanning tree is



(2) Breadth first search (BFS)

In this algorithm a rooted tree will be constructed, and the underlying undirected graph of this rooted forms the spanning tree. The idea of BFS is to visit all vertices on a given level before going to the next level.

Procedure:

Step 1:

Choose a vertex and designate it as the root. Then add all edges incident to this vertex, such that the addition of edges does not produce any cycle.

Step 2:

The new vertices added at this stage become the vertices at level 1 in the spanning tree, arbitarily order them.

Step 3:

Next, for each vertex at level 1, visited inorder, add each edge incident to this vertex to the tree as long as it doesnot produce any cycle.

Step 4:

Arbitarily order the children of each vertex at level 1. This produces 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:

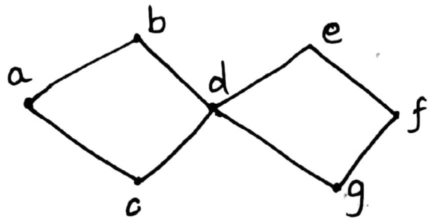
The procedure ends, since there are only a finite number of edges in the graph.

Step 7:

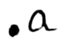
A spanning tree is produced since we have produced a tree without cycle containing every vertex of the graph.

example:

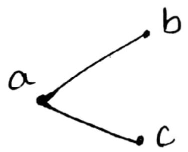
(1) Obtain the spanning tree of a following graph using BFS algorithm



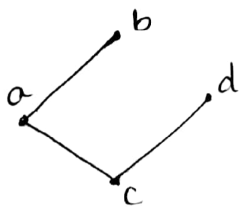
(i) Choose the vertex 'a' to be the root



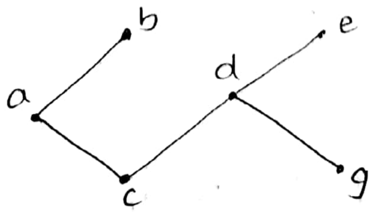
(ii) Add edges incident with all vertices adjacent to 'a', so that edges $\{a,b\}$, $\{a,c\}$ are added.
The two vertices b and c are on level 1 in the tree



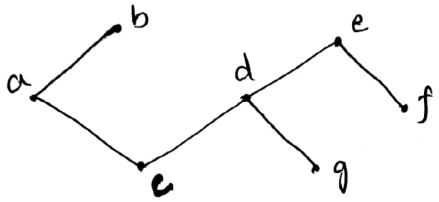
(iii) Add edges from these vertices at level 1 to adjacent vertices not already in the tree. Hence the edge $\{c,d\}$ is added. The vertex d is in level 2.



(iv) Add edge from d in level 2 to adjacent vertices, not already in the tree the edges $\{d,e\}$ and $\{d,g\}$ are added. Hence e and g are in level 3.



(v) Add edge from e at level 3 to adjacent vertices not already in the tree and hence {e,f} is added.



This is required BFS.

* Spanning Trees:

Minimal spanning trees:

A minimal spanning tree G is a spanning tree in which the sum of the weights associated with all edges in it is a minimum.

Algorithm for minimal spanning trees:

There are several methods available for actually finding a minimal spanning tree in a given graph.

To find a minimal spanning tree for a connected weighted graph, there are 2 algorithms.

(1) Kruskal's Algorithm

(2) prim's Algorithm

(1) Kruskal's Algorithm:

The working rule of the kruskal's method (usually called kruskal's algorithm) may be stated as follows.

Step 1:

Given a connected, weighted graph G with n vertices, list the edges of G in the order of non-decreasing weights.

Step 2:

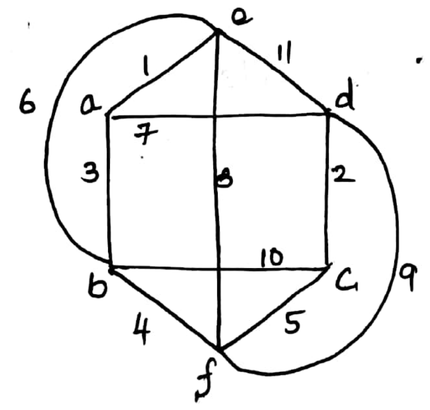
Starting with a smallest weighted edge, proceed sequentually by selecting one edge at a time such that no cycle is form.

Step3 :

Stop the process of step 2, when n-1 edges are selected. These n-1 edges constitute a minimal spanning tree

* Problems

(1) Find The minimal spanning tree for the following graph using Kruskal's algorithm.



Weighted graph (G)

This graph contains 6 vertices and thus we obtain a spanning tree with 5 edges. Now, we arrange the edges of the graph in increasing order of their weights

Weights of edges

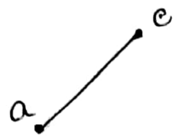
Step1: List the edges in non-decreasing order of their weights
(increasing written above "non-decreasing")

edge	(a,e)	(c,d)	(a,b)	(b,f)	(e,f)	(b,e)	(a,d)	(e,f)
Weight	1	2	3	4	5	6	7	8

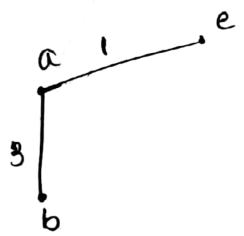
(d,f)	(b,c)	(e,d)
9	10	11

Now, a minimal spanning tree is constructed as

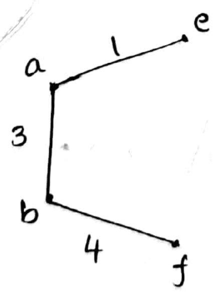
Step 2: Select the edge (a,e) since it has the smallest weight, include it in T.



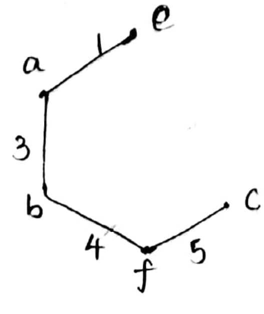
Step 3: select an edge with the next smallest weight (a,b) since it doesnot form cycle with the existing edges in T, so include it in T.



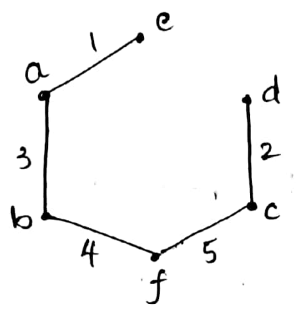
Step 4: select an edge with the next smallest weight (b,f) since it doesnot form cycle with the existing edges in T. So include it in T.



Step 5: select an edge with the next smallest weight (c,f) since it doesnot form cycle with existing edges in T, so include it in T.



Step6 : select an edge with the next smallest weight (c,d) since it doesnot form cycle with the existing edges in T, so include it in T.

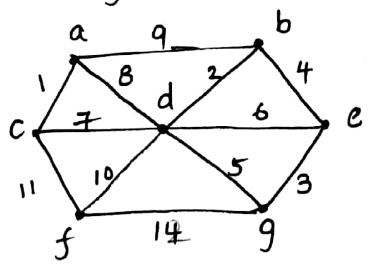


since G contain 6 vertices and we have choosen 5 edges. we stop the algorithm and the minimal spanning tree is produced.

Weights of the minimal spanning tree is

$1+2+3+4+5 = 15$ units

(2) Find The minimal spanning tree of The following graph using kruskal's algorithm



The graph consists of 7 vertices and we obtain a spanning tree with 6 edges.

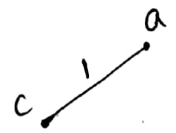
Step1 : List the edges in non decreasing order of their weights

edges	(a,c)	(b,d)	(e,g)	(b,e)	(d,g)	(d,e)	(d,c)	(a,d)
Weight	1	2	3	4	5	6	7	8

(a,b)	(d,f)	(c,f)	(f,g)
9	10	11	12

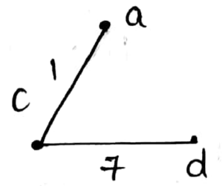
Step 2 :

select the edge (a,c) since it has the smallest weight, include it in T.

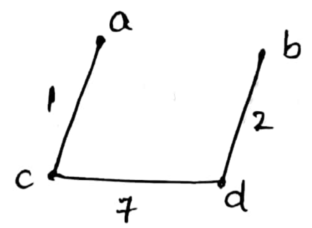


Step 3 :

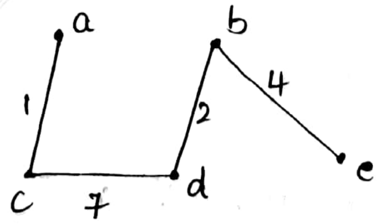
select the edge (c,d) with the next smallest weight, since it doesnot form cycle with the existing edges in T, so include it in T.



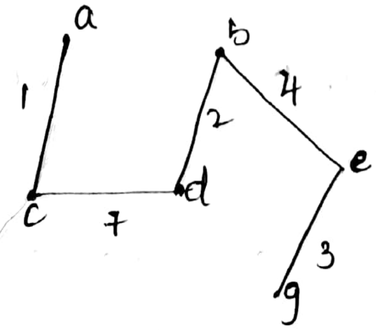
Step 4 : select the edge (d,b)



Step 5 : select an edge (b,e)

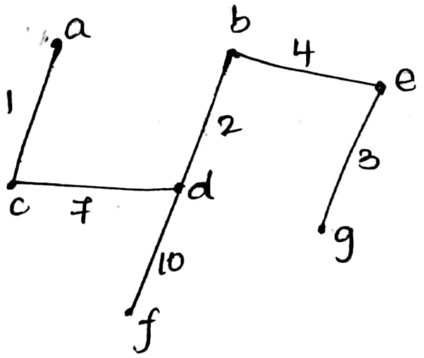


Step 6 : select an edge (e,g)



Step7 :

select an edge (d,f)



Since G contains 7 vertices and we obtain 6 edges, we stop the algorithm and the minimal spanning tree is produced.

The weight of minimal spanning tree is

$1+2+3+4+7+10 = 27$