SRM IST RAMAPURAM

DEPARTMENT OF MATHEMATICS

Sub. Code: 18MAB302T

Sub. Tittle: Discrete Mathematics for Engineers

Module : 6

graph theory

Introduction:

Graph:- A graph $G = \langle V, E, \phi \rangle$ consists of a non-empty set V called set of vertices (or nodes or points) set E is said to be the set of edges of the graphs, E is said to be the set of edges and ϕ is mapping from the set E to a set of ordered pair of elements of V.

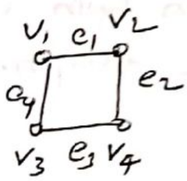
ie, $\phi : E \to V \times V$.

Note:- If an edge ~~e∈G~~ $e \in E(G)$ is associated with an ordered pair (u, v) then e is said to connect or join the node u and v.

The edge $e \in E$ is said to be incident with the vertices u and v.

Also if $e = (u, v) \in E$ then we say u is adjacent with v.

Example: G_1:



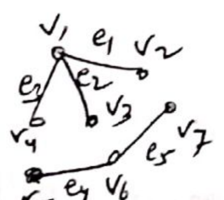
$G_1 = (V_1, E_1)$

where $V_1 = \{v_1, v_2, v_3, v_4\}$

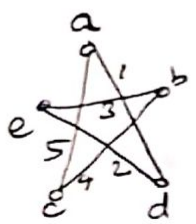
$E_1 = \{e_1 = (v_1, v_2), e_2 = (v_2, v_4)$

$e_3 = (v_3, v_4), e_4 = (v_3, v_1)\}$

G_2:



G_3:

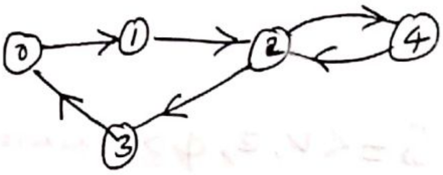


$V = \{a, b, c, d, e\}$

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

Directed graph: A graph with every edge is directed (One way (or) two ways).

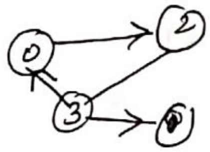
Ex: G :



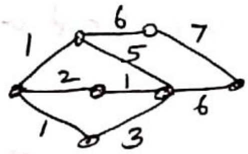
Undirected graph: A graph with every edge is undirected.

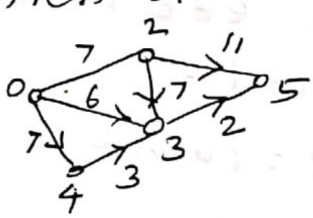
mixed graph: A graph with some edges are directed and rest are not.

Ex:

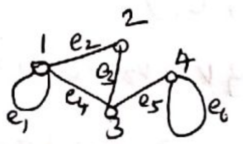


weighted graph: A graph with each edges (or) each vertices (or) both given a numerical value.

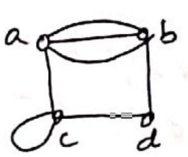




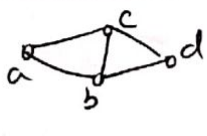
multiple graph: A graph with loops are called multiple graph.



pseudo graph: A graph with loops and multiple edges are allowed.



Simple graph: A graph without self loop and multiple edges.



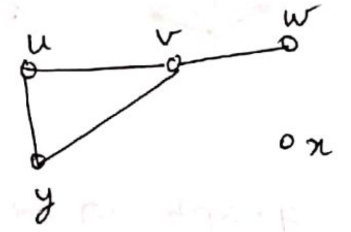
Remark: most of the graph there we considered are simple graph.

Degree of a vertex: A degree of a vertex (say v) in a graph is the number of vertices which are adjacent to v. denoted by $d_G(v)$ or $d(v)$

$d(v)$ = no. of edges which are incident to v.

A vertex with 0 degree called isolated vertex and a vertex with 1 degree called pendant vertex.

Ex:



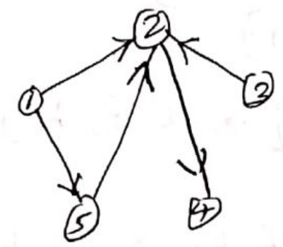
degree of $u = d(u) = 2$
$d(v) = 3$
$d(w) = 1$
$d(y) = 2$
$d(x) = 0$

x ~ isolated vertex
w ~ pendant vertex.

Degree of a vertex in Directed graph:

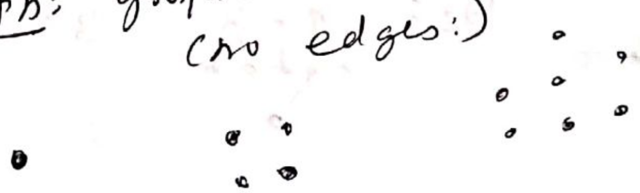
* Indegree of v: The number of adjacent vertices which is coming towards v.

* Out degree of v: The number of adjacent vertices which is going out from v.



vertices:	1	2	3	4	5
In degree:	0	3	0	1	1
out degree:	2	1	1	0	1

Null graph: Graph contain only isolated vertices. (no edges.)



order of a graph $G = O(G) = n$ = no. of vertices of G

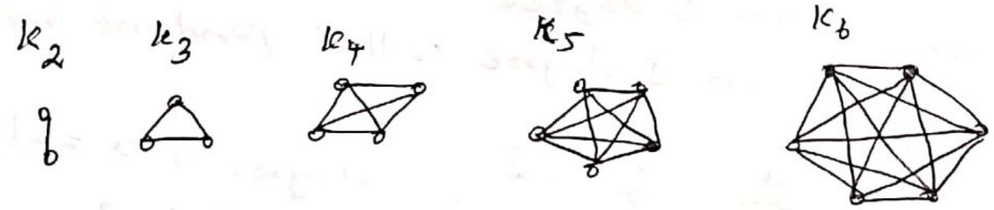
Size of a graph $G = m$ = no. of edges of G

Scanned by CamScanner

Complete graph: A simple graph with each pair of vertices are connected by an edge.

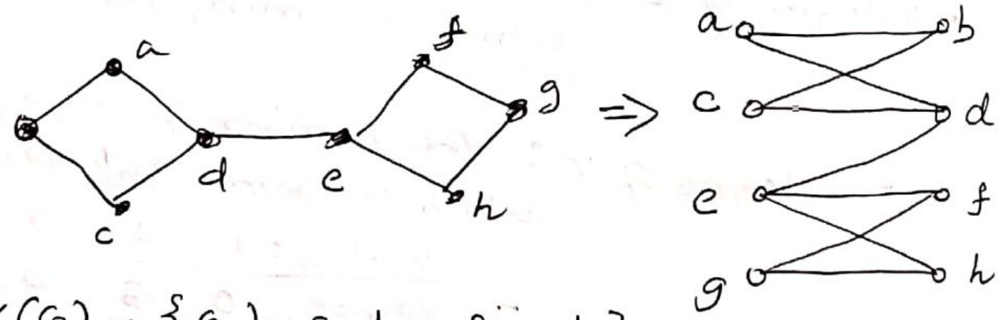
(or) A graph with every vertex is connected to every other vertices.

Denote K_n : Complete graph with n vertices.



Bipartite graph: A graph G whose vertex set V can be divided into two disjoint and indepedent sets V_1 and V_2 such that each edge e in E(G) connect a vertex in V_1 ~~and~~ to a vertex in V_2.

G:



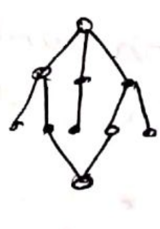
$v(G) = \{a, b, c, d, e, f, g, h\}$

$V_1 = \{a, c, e, g\}$, $V_2 = \{b, d, f, h\}$

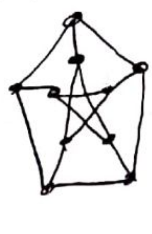
check the following are bipartite or not?

G_1 :

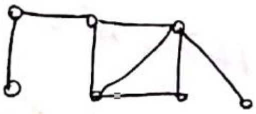
G_2 :



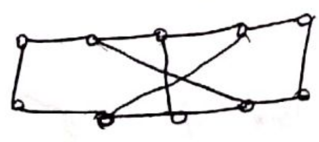
G_3 :



G_4 :

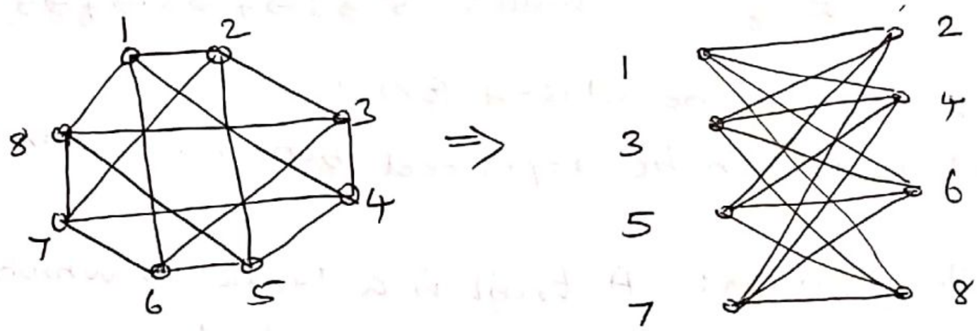


G_5 :



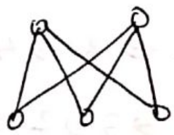
<u>Complete bipartite graph:</u> A bipartite graph G whose vertex set V can be divided into two disjoint and independent sets V_1 and V_2 such that every vertex of V_1 is adjacent with every other vertices of V_2.

Denote it by $K_{p,q}$ where $p = |V_1|$ & $q = |V_2|$



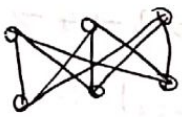
$K_{4,4}$.

$K_{2,3}$:



$K_{2,4}$:

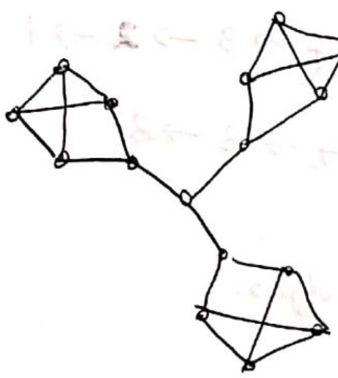
$K_{3,3}$:



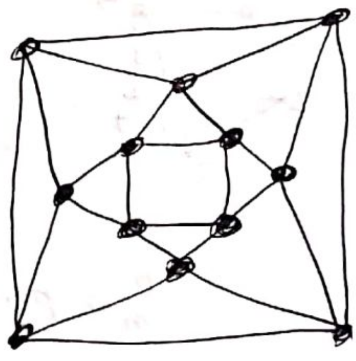
$K_{1,4}$:

<u>Regular graph:</u> A graph with all the vertices of same degree.

if $d_G(v) = r$ for all $v \in V(G)$ then G is said to be r-regular graph.



3-regular graph



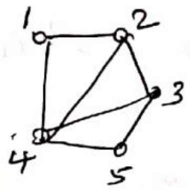
4-regular graph.

Ex: Discribe the 2-regulargraph:-

Walk of a graph: A walk is a sequence of vertices and edges of a graph.

ie, if we traverse a graph then we get a walk.

Ex:



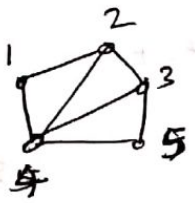
walk: $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 3$

walk: $5 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 5$

Note: Walk can be closed (or) open

Note: walk can be repeated both vertices and edges

Trail of a graph: A trail is a walk in which no edges is repeated. (vertices may repeat).



$Trail_1$: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2$

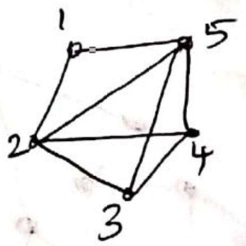
$Trail_2$: $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$

$Trail_3$: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 1$

Circuit of a graph: A closed trail is called the circuit.

Example: $Trail_3$ on above graph is a circuit.

path of a graph: A path is a trail in which ~~its neight may~~ neither the vertices nor the edges are repeated.

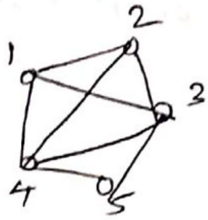


P_4 : $5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$

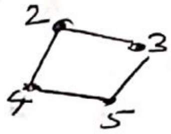
P_3 : $2 \rightarrow 4 \rightarrow 3 \rightarrow 2$

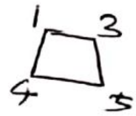
P_n : path on n edges.

Cycle: A closed path is a cycle.

G: 

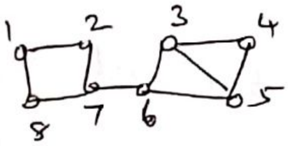
C_3: 

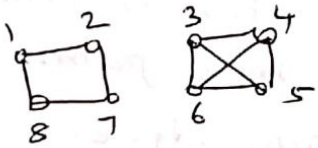
C_4: 

C_4: 

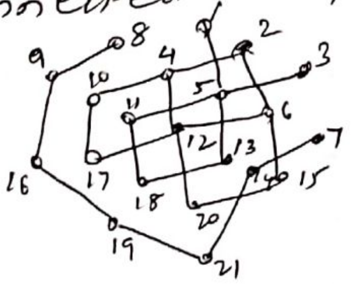
Connected graph: A graph G is said to be connected if every vertices of G is connected to every other vertices of G by a path.

otherwise G is said to be disconnected.

Example: G :  – connected

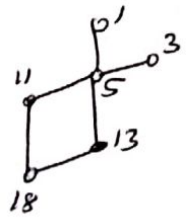
G :  – disconnected.

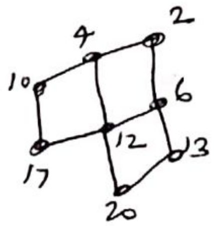
If a graph G is disconnected then it has disconnected component.

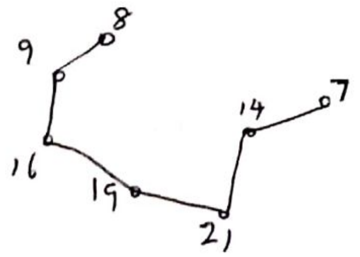
G : 

$\Rightarrow G = \langle G_1, G_2, G_3 \rangle$

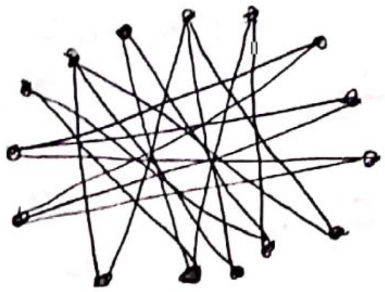
G_i – disconnected component of G.

G_1 : 

G_2 : 

G_3 : 

1. Is this graph is disconnected? if so, find the no. of disconnected component.



2. Explain three kind of traversal with an example.

3. Draw all the graph with
(i) 3 vertices (ii) 4 vertices (iii) 5 vertices
(iv) 6 vertices

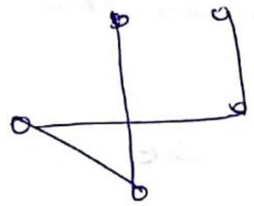
Tree : A connected and acyclic graph.
ie, Every vertex is connect to every other vertex by a path. and it doesnot contain ~~a cyc~~ any cycle.

Scanned by CamScanner

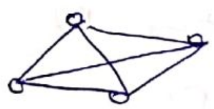
Degree Sequence of a graph.

Degree Sequence of a graph is the list ~~all the~~ of degrees of all the vertices of the graph. Usually we list them in from nonincreasing order, that is ~~the~~ largest to smallest degree.

Ex:

G_1  : 2, 2, 2, 1, 1.

G_2  : 2, 2, 2, 1, 1,

G_3:  : 3, 3, 3, 3

Theorem 1: Fundamental theorem on GT (or Handshaking theorem.)

Statement:- In any graph G, the sum of the degree of its vertices is equal to the twice the number of edges.

i.e. $\sum_{i=1}^{n} d(v_i) = 2e$ where n - no. of vertices of G and e - no. of edges of G.

Proof:

Let G be a graph with e-edges and n-vertices

Let $V(G) = \{v_1, v_2, \ldots, v_n\}$

$E(G) = \{e_1, e_2, \ldots, e_e\}$

Since each edge e_j contributes 2 degrees, the sum of the degrees of all vertices in G is equal to twices the number of edges.

i.e., $\sum_{i=1}^{n} d(v_i) = 2e.$

~~Theorem~~:

Corollary 1: The number of odd degree vertices in a graph is even.

Proof:

Let $G = (V, E)$ be a graph.

Let ~~V_1 = no. of~~

$V_1 \subseteq V$ consist of all odd degree vertices

& $V_2 \subseteq V$ " even "

By theorem 1.

we have $\sum_{i=1}^{n} d(v_i) = 2e$

$\Rightarrow \sum_{v \in V_1} d(v) + \sum_{v \in V_2} d(v) = 2e = \text{even}$

$$\Rightarrow \sum_{v \in V_1} d(v) = 2e - \sum_{v \in V_2} d(v)$$

[note $\sum_{v \in V_2} d(v) = \text{even}$ ∵ each $d(v)$ - even]

$$\Rightarrow \sum_{v \in V_1} d(v) = \text{even} - \text{even}$$

$$= \text{even}$$

$$\Rightarrow \sum_{v \in V_1} d(v) = \text{even}$$

$$\Rightarrow |V_1| = \text{even} \quad ∵ d(v) - \text{odd}.$$

Ex:

1. Is the following are degree sequence (graphical sequence) of a graph. so then Draw the graphs

(a) 3,3,3,3 (b) 2,2,1,1,1,1 (c) 5,3,2,1,1,1,1

(d) 5,3,2,1,1 (e) 2,2,2,2,2,2,2,2 (f) 7,6,5,4,3,2,1

2. How many vertices will the following graph contain

(i) 16 edges and all vertices of degree 2.

(ii) 21 edges, 3 vertices of degree 4 and other vertices of degree 3.

(iii) 24 edge and all vertices of same degree

Matrix Representation of graphs

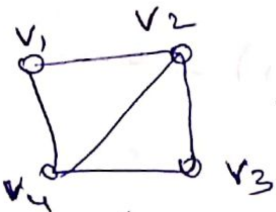
Adjacency matrix:

Let G be a simple graph with n vertices (say $v_1, v_2, \dots, v_n$) the matrix A (or) $A_G = [a_{ij}]$ of order $n \times n$

where $a_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is adjacent with } v_j \\ 0, & \text{otherwise.} \end{cases}$

Here A – adjacency matrix of G.

Example:



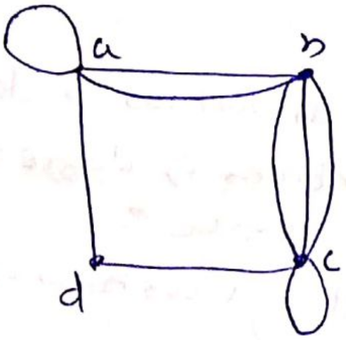
G

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

For **pseudo graph:**

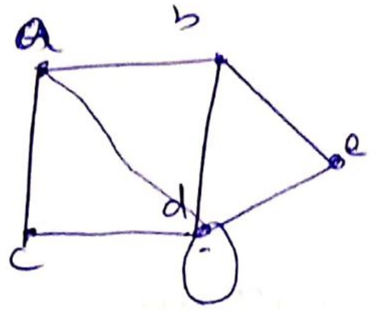
$A_G = [a_{ij}]$ where $a_{ij} = \begin{cases} 2 & \text{if } v_i \text{ forming loop} \\ k & \text{if } k \text{ multiple edges between } v_i \text{ \& } v_j \\ 0 & \text{otherwise} \end{cases}$

G.



$$A_G = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

multigraph (loops only allowed)

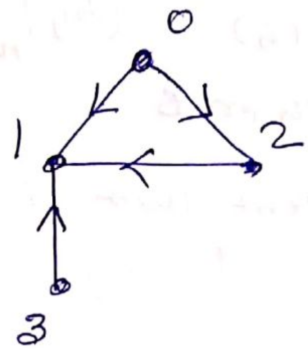


$$A_G: \begin{array}{c|ccccc} & a & b & c & d & e \\ \hline a & 0 & 1 & 1 & 1 & 0 \\ b & 1 & 0 & 0 & 1 & 1 \\ c & 1 & 0 & 0 & 1 & 0 \\ d & 1 & 1 & 1 & 1 & 1 \\ e & 0 & 1 & 0 & 1 & 0 \end{array}$$

Directed graph:

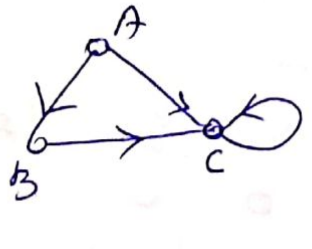
$$A_G = [a_{ij}] \qquad a_{ij} = \begin{cases} 1 & \text{if } v_i \longrightarrow v_j \\ 0 & \text{otherwise} \end{cases}$$

G_1



$$A_{G_1}: \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \end{array}$$

G_2:



$$A_{G_2}: \begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 1 & 1 \\ B & 0 & 0 & 1 \\ C & 0 & 0 & 1 \end{array}$$

Remarks:

In simple graph: G

(i) A_G is symmetric.

(ii) $a_{ii} = 0$

(iii) degree v_i = sum of i^{th} row.

Incidence matrix:

Let $G = (V, E)$ be graph (simple).

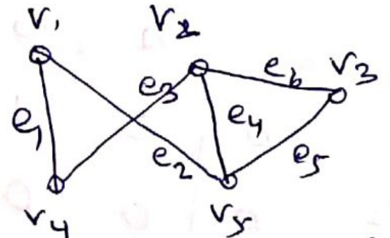
with $|V(G)| = n$ and $|E(G)| = m$

The incidence matrix $B(G) = [b_{ij}]_{n \times m}$ (or) B_G (or) B

where

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

Ex:



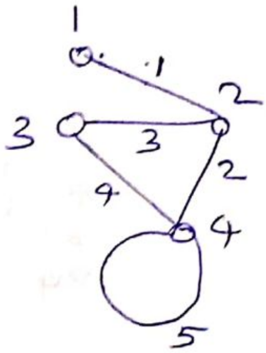
$$B(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

pseudo graph (both loop & multiple edges allowed)

incidence matrix $= B_G = [b_{ij}]$

$$b_{ij} = \begin{cases} 1 & \text{if } e_j \text{ is incident with } v_i \\ 2 & \text{if } e_j \text{ is a loop at } v_i \\ 0 & \text{otherwise} \end{cases}$$

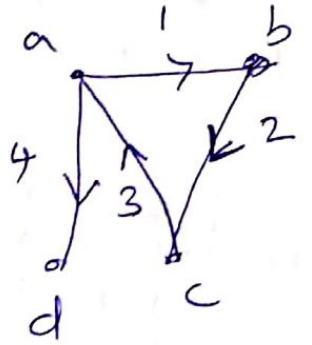
Ex:



$$B_G = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 \end{bmatrix} \end{array}$$

Directed graph:

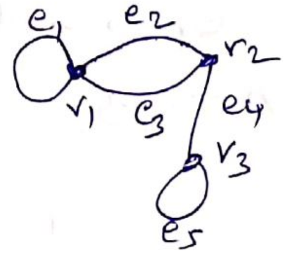
$$b_{ij} = \begin{cases} -1 & \text{if } e_j \text{ is leaving from } v_i \\ 1 & \text{if } e_j \text{ is enters in } v_i \\ 0 & \text{otherwise} \end{cases}$$

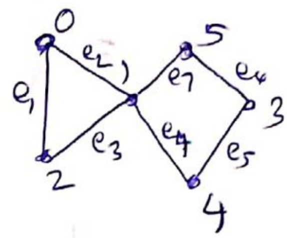


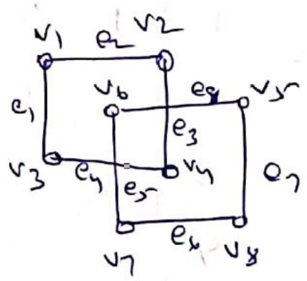
$$B_G = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

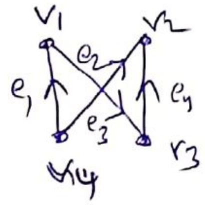
Excercise!

(1) Write the Adjacency and incidence matrix of the following graph.

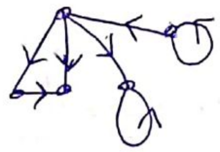
(a) G_1: 

G_2: 

G_3: 

G_4: 

G_5: 

G_6: 

(2) Draw the graph associated with the following matrix (Adjacency matrix)

(i) $\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ (ii) $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 1 \\ 0 & 0 & 1 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

(ii) $\begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ (iii) $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

(3) Draw the graph for the following incidence matrix.

(i)
$$\begin{array}{c} A \\ B \\ c \\ d \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

(ii)
$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

(iii)
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$$

(3) Draw a multigraph whose adjacency matrix is

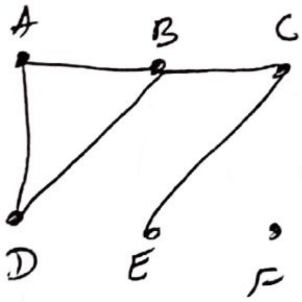
$$A(G) = \begin{bmatrix} 1 & 3 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 & 2 \\ 0 & 1 & 1 & 2 & 0 \end{bmatrix}$$

Adjacency list:

For a given graph G, the adjacency list representation is given by their adjacency vertex set.

Ex:

G:



vertex	Adjacency list
A	B, D
B	A, C, D
C	B, E
D	A, B
E	C
F	ϕ

Graph isomorphism:

Two graph G_1 and G_2 are said to be isomorphic if

(i) $|V(G_1)| = |V(G_2)|$ = no. of vertices are equal

(ii) $|E(G_1)| = |E(G_2)|$ = no. of edges are equal.

(iii) Degree sequence of G_1 = DS (G_2)

(iv) iff Adjacency preserves.

i.e., There exist a function 'f' from $V(G_1)$ to $V(G_2)$

i.e. $f : V(G_1) \to V(G_2)$

(i) f is one-one and onto.

(ii) f is preserves adjacency.

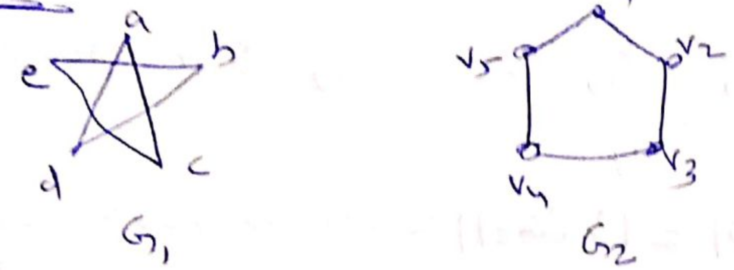
i.e. if $(u,v) \in E(G_1)$
then $(f(u), f(v)) \in E(G_2)$.

Note:

(i) if $G_1 \cong G_2$ then $A_{G_1} = A_{G_2}$ in some order.

Scanned by CamScanner

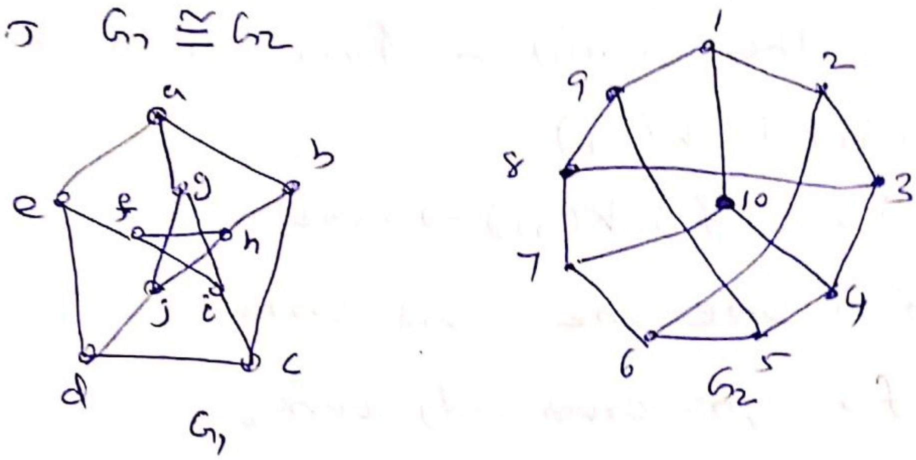
Example:

(1) 

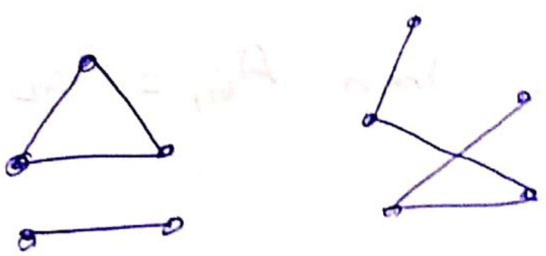
$f(a) = v_1$, $f(b) = v_4$, $f(c) = v_2$, $f(d) = v_5$

$f(e) = v_3$

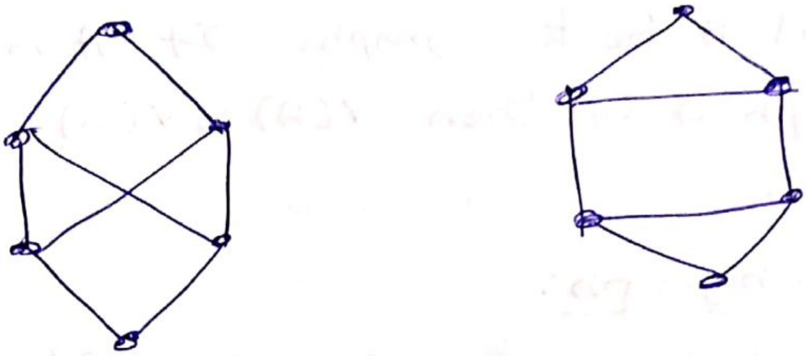
(2) P.T $G_1 \cong G_2$



(3) If $G_1 \cong G_2$?



15) Is $G_1 \cong G_2$?



Scanned by CamScanner

Subgraphs:

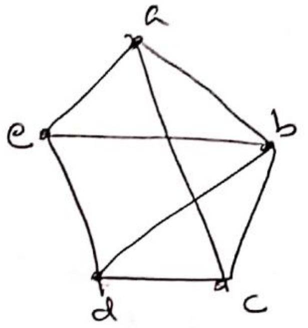
Let G and H be two graphs. If H is called the Subgraph of G then $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Spanning subgraph:

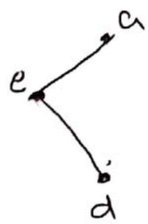
Let G and H be two graphs. If H is said to be a spanning subgraph G then $V(H) = V(G)$ and $E(H) \subseteq E(G)$.

Example:

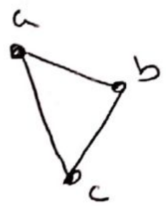
Given G :



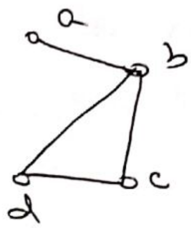
H_7 :



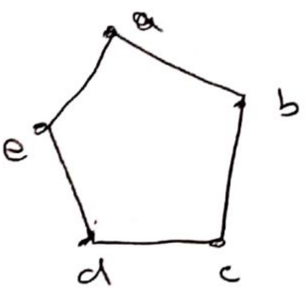
H_1 :



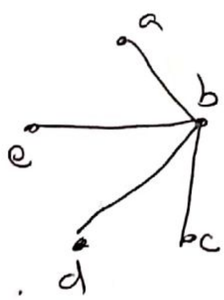
H_2 :



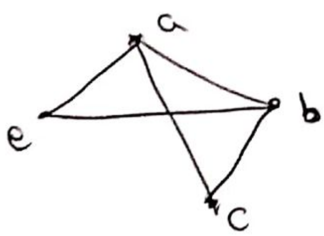
H_3 :



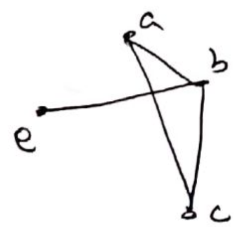
H_4 :



H_5 :



H_6 :



Here (i) $H_i \; \forall \; i = 1,2,3,4,5,6,7$ are subgraph of G

(ii) H_3 & H_4 are the only spanning subgraph.

(iii) H_1 & H_5 are the only induced subgraph

~~(iv) H_7 is edge disjo~~

If two subgraph say H_i & H_j are edge disjoint subgraph of G then there is no edge is common in H_i & H_j.

Example: above example

(i) H_1 & H_7

(ii) H_2 & H_7

(iii) H_4 & H_7

(iv) H_6 & H_7

Connectness:

Connected: There exist a path between any pair of vertices.

Theorem: If G is a simple graph with n vertices and k-components then it can have atleast $\frac{(n-k)(n-k+1)}{2}$ edges.

Proof: Let G be a simple graph with n vertices and k-components. Let $G_1, G_2, \ldots G_k$ be its component.

Let $n_1, n_2, \ldots, n_k$ be the number of vertices of $G_1, G_2, \ldots, G_k$ respy.

Note $n = \sum_{i=1}^{k} n_i$

Since each G_i, is a simple connected graph. The maximum number of edges in each G_i is $\frac{n_i(n_i-1)}{2}$, ie., $|E(G_i)| \leq \frac{n_i(n_i-1)}{2}$ — ①

observe that, $|E(G)| = \sum_{i=1}^{k} |E(G_i)|$

by ① $\Rightarrow |E(G)| \leq \sum_{i=1}^{k} \frac{n_i(n_i-1)}{2}$

Note that $n_i \leq n-(k-1)$ {even if the remaining $k-1$ components are isolated vertices}.

Therefore,

$$|E(G)| \leq \sum_{i=1}^{k} \frac{n_i(n_i-1)}{2} \leq \sum_{i=1}^{k} \frac{(n-k+1)(n_i-1)}{2}$$

$$\Rightarrow |E(G)| \leq \frac{(n-k+1)}{2} \sum_{i=1}^{k} (n_i-1)$$

$$\Rightarrow |E(G)| \leq \frac{(n-k+1)(n-k)}{2}$$

$$\left[\because \sum_{i=1}^{k} (n_i-1) = \sum_{i=1}^{k} n_i - k\right]$$

$$\therefore |E(G)| \leq \frac{(n-k)(n-k+1)}{2}.$$

Operations on Graph:

1. **Union of graphs:** Given two graphs (Input)

~~Let~~ **Input:** $G_1 = (V_1, E_1)$ & $G_2 = (V_2, E_2)$

Output: Union of G_1 and G_2

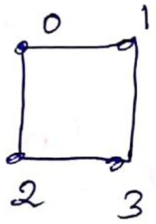
$$G_1 \cup G_2 = G$$

such that $V(G) = V(G_1) \cup V(G_2)$

$E(G) = E(G_1) \cup E(G_2)$

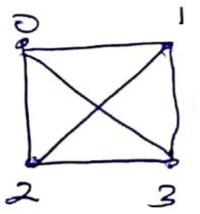
Example:- 1

G_1:



G_2:

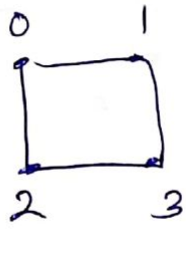
$G = G_1 \cup G_2$:



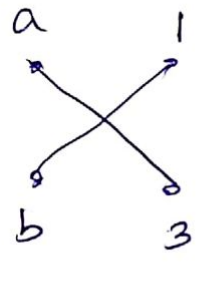
: K_4

Ex: 2

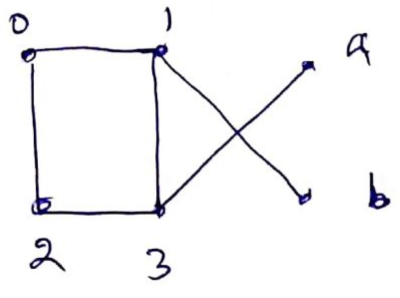
G_1:



G_2:



$G = G_1 \cup G_2$:

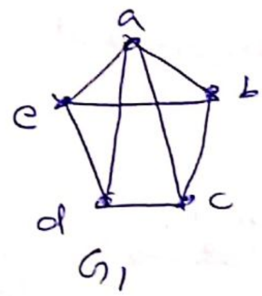


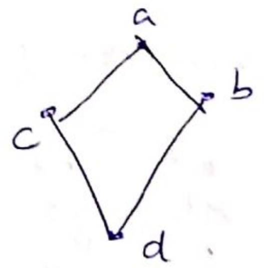
2. Intersection :

Input : Two graphs : $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$

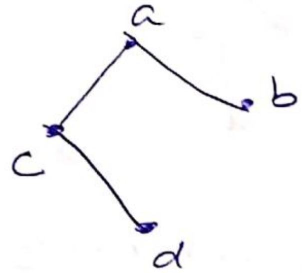
output : Intersection of G_1 & $G_2 = G_1 \cap G_2 = G$
s.t $G = (V, E)$ where
$V(G) = V(G_1) \cap V(G_2)$ & $E(G) = E(G_1) \cap E(G_2)$

Example:





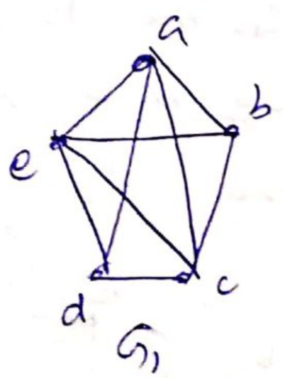
$G = G_1 \cap G_2$:

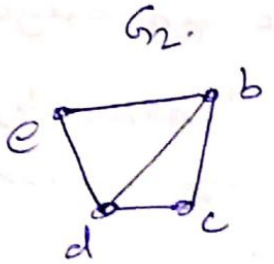


3. Ring Sum :

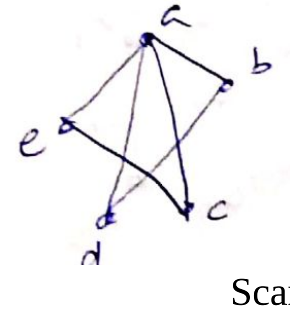
The Ring sum of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a simple graph, denoted by $G = G_1 \oplus G_2$ with $V(G) = V(G_1) \cup V(G_2)$ and the edges of G are the edges either in G_1 or G_2 but not in both.

Ex :



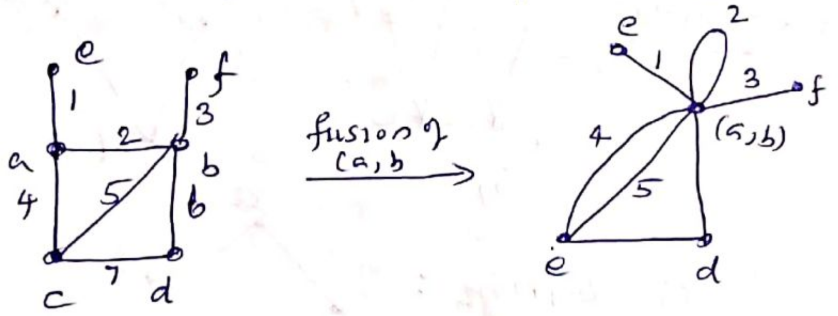


$G = G_1 \oplus G_2$:



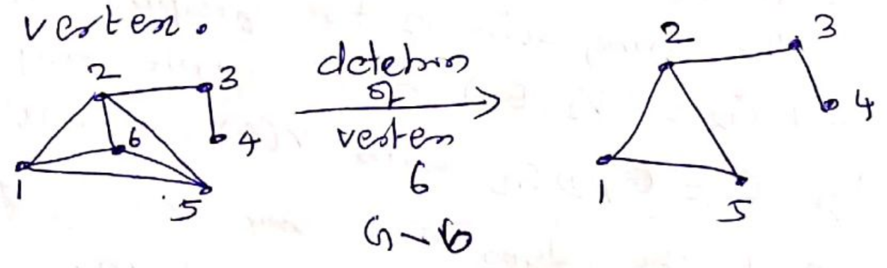
4. <u>Fusion of graph</u>: A pair of vertices (a, b) in a graph is said to be fusion (or merged (or) identified) if the two vertices are replaced by a single new vertex such that every edge that was incident on either 'a' (or) 'b' (or) on both is incident on the new vertex. [~~contraction of (a,b)~~]

<u>Example:</u>



5. <u>Deletion of a vertex</u>

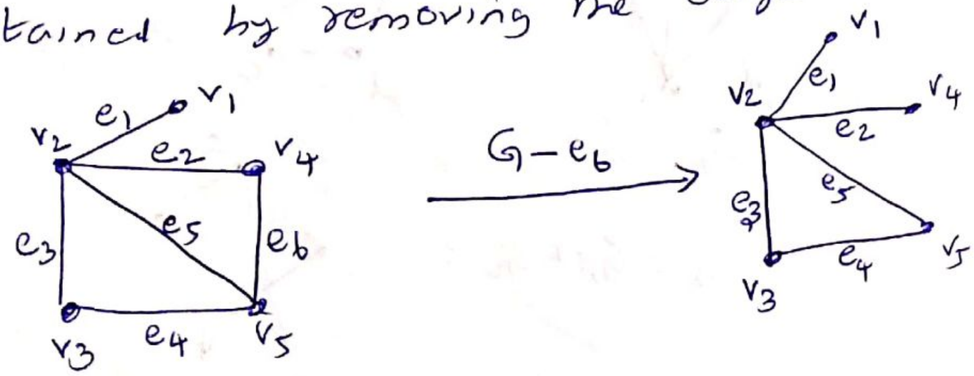
Deletion of a vertex in a graph implies that deletion of all its incident edges to its vertex.



6. <u>Deletion of an edge:</u>

Let 'e' be an edge of G. The deletion of 'e' from G is denoted by G−e is a graph obtained by removing the edge e alone.

Ex:



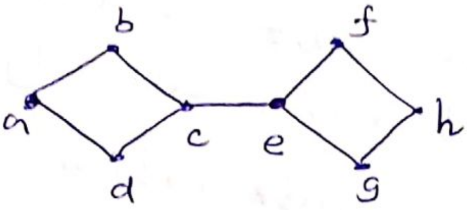
cut vertex: Let G be a connected graph. A vertex $v \in G$ is called a cut vertex of G, if $G-v$ results in disconnected graph.

Note: 1. A connected graph G may have at most $(n-2)$ cut vertices, where $n = |V(G)|$

2. $G-v$ may have atmost $n-1$ disconnected components.

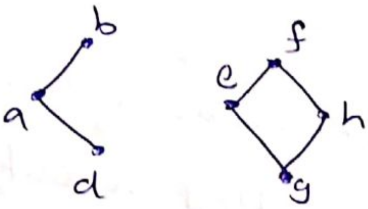
Example:

G:

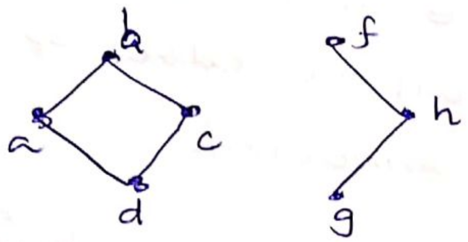


cut vertices of G are c and e.

$G-c$:



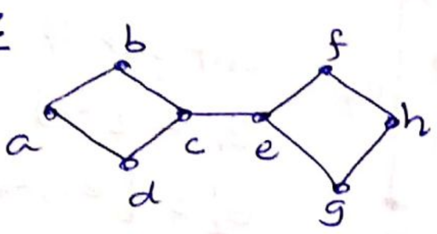
$G-e$:



cut edge: Let G be a connected graph. An edge 'e' is called cut edge of G, ~~denoted by $G-e$~~ if $G-e$ results in disconnected.

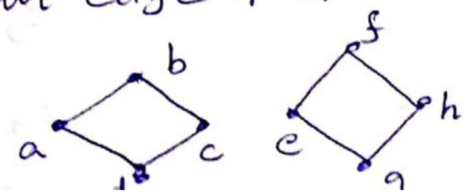
Example:

G:



Here (c,e) is a cut edge of G

$G-(c,e)$:



Note: Let 'G' be a connected graph with 'n' vertices.

* a cut edge $e \in G$ iff the edge 'e' is not a part of any cycle in G.

* The maximum number of cut edge possible is 'n-1'

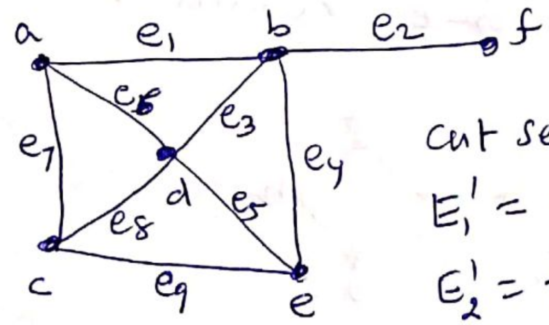
* Whenever cut edges exist, cut vertices also exist. because at least one vertex of a cut edge is a cut vertex.

* If a cut vertex exist then the cut edge may (or) may not exist.

Cut set: Let $G=(V,E)$ be a connected graph. A subset E' of E is called a cut set of G if deletion of all the edges of E' from G makes G is disconnected.

Observation: If deleting a certain number of edges from a graph makes it disconnected then those deleted edges are called the cut set of the graph.

Ex: G :



cut set of G are

$E_1' = \{e_2\}$

$E_2' = \{e_1, e_6, e_8, e_9\}$

$E_3' = \{e_1, e_3, e_4\}$

$E_4' = \{e_1, e_3, e_8, e_9, e_5\}$

Theorem: prove that every circuit has an even number of edges in common with any edge cut.

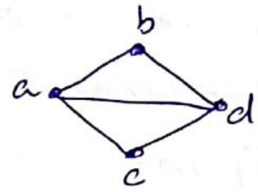
proof:- Consider an edge cut S in a graph G. Let the removal of S partition $V(G)$ into two disjoint subsets V_1 and V_2. Let C be a circuit in G. The initial and final vertex u of C is contained in either V_1 (or) V_2, say $u \in V_1$. Starting from u and traversing along C, we go equally often from V_1 to V_2 along an edge in C as from V_2 to V_1, since we start and end at u.

Because every edge with one endvertex in V_1 and one in V_2 is contained in S.

Hence the theorem.

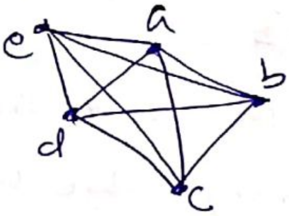
Eulerian Graph:

A path of a graph G is called an Eulerian path, if it includes each edge of G exactly once.

Example: G : 

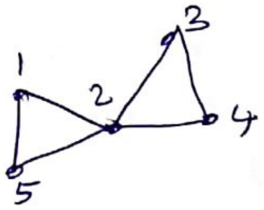
Euler path: a b d c a d

An Euler circuit is a circuit that uses every edge of a graph exactly once.

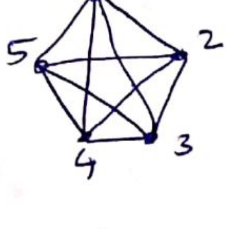
Example: G : 

Euler circuit: a b c d e b d a c e a

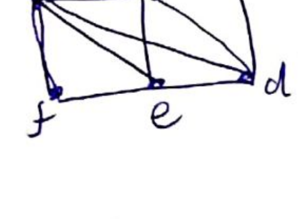
A graph contains an Euler circuit is called Eulerian graph.

Example: G : 

EC : 1 2 3 4 2 5 1

G_2 : 

EC : 1 2 3 4 5 1 3 5 2 4 1

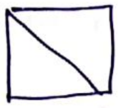
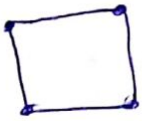
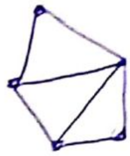
G_3 : 

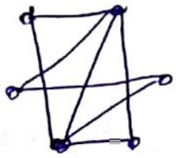
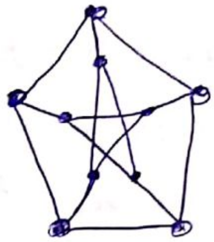
EC : a b c d e f a e b d a.

Scanned by CamScanner

Exercise:

1. verify the following graph is Eulerian or not?

(i) 
(ii) 
(iii) 

(iv) 
(v) 
(vi) 

Theorem:- (Necessary and sufficient condition for Eulerian).

A connected graph G is Eulerian if and only if every vertex of G is even degree.

Proof:-

Assumme: G is Eulerian.

To prove that: every vertex of G is even degree.

Given G is Eulerian ⇒ G must have Eulerian cycle, say C.

Each time the cycle C passes through a vertex it contributes two to the vertex's degree, except the starting and ending vertices. If the ~~path~~ cycle terminates where it started, it will contribute two to the degree as well.

Hence every vertex has even degree.

Assumme: Every vertex of G is even degree

To prove that: G is Eulerian.

Given G is having even degree for every vertex. To prove G is Eulerian, we must construct an Eulerian circuit.

We start with a proper vertex and construct a cycle. We delete the edge that we visited every time. That is if we visit a vertex then its degree is reduce by two. If this cycle contains all edges of the graph, then stop.

Otherwise, select a vertex of degree greater than 0 (that belongs to the graph as well as to the cycle!) and construct another cycle. Then splice these two cycles into one.

If a join cycle contains all the edges of the graph then stop.

If not, select a common vertex of degree greater than 0 and construct a cycle, and so on until all the edges are covered.

Thus G has eulerian cycle.

$\Rightarrow$ G is Eulerian.

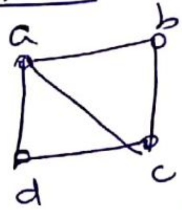
Hamiltonian Graph:

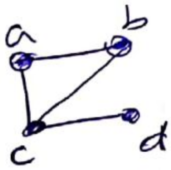
* A path of a graph G is called *Hamiltonian path* if it includes each vertex of G exactly once.

* A cycle of a graph G is called a *hamiltonian cycle* if it includes each vertex of G exactly once except starting and ending vertex appears twice.

* A graph containing a hamiltonian cycle is called a *hamiltonian graph.*

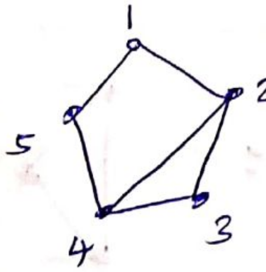
Example:

G_1:  — Hamiltonian [HC: a b c d a]

G_2:  — non-hamiltonian. [no hamiltonian cycle]

There is a hamiltonian path

HP: d c b a

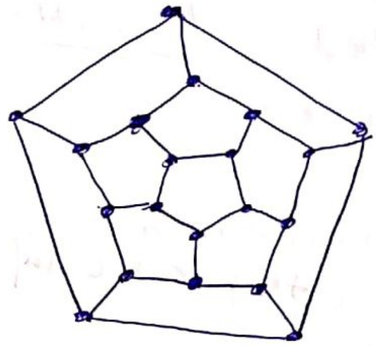
G_3:  — Hamiltonian graph.

HC: 1 2 3 4 5 1

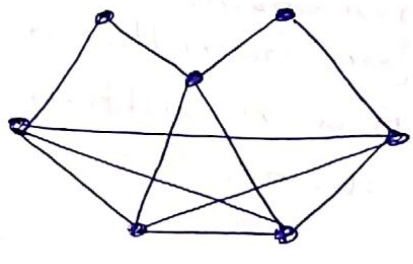
Ex:

(i) Check the following graph is Eulerian (or) hamiltonian (or) both. If it is Eulerian (or) Hamiltonian (or) both, then find its Eulerian circuit (or), hamiltonian cycle (or) both. If its not Eulerian (or) hamiltonian (or) both then find it Eulerian path (or) Hamiltonian path (or) both (if exist). Justify.

G_1:

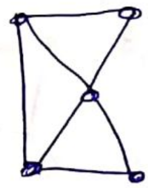


G_2:

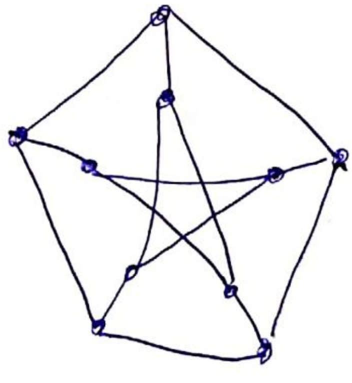


G_3:

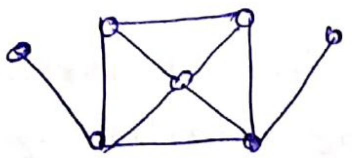
G_4:



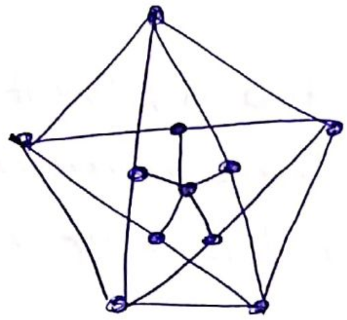
G_5:



G_6 :



G_7 :



planar graph:

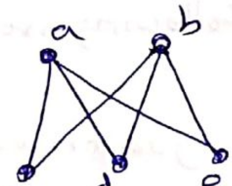
A graph is said to be planar if it can be drawn in a plane so that no edge cross.

When connected graph can be drawn without any edges crossing, it is called planar.

When planar graph is drawn in this way, it divides the plane into regions called faces.

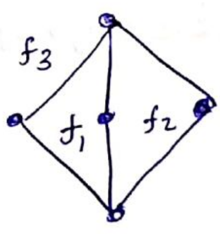
Example:

$G \cong K_{2,3}$:



is a planar graph

$\Downarrow$ planar drawing



f_1, f_2 - interior faces

f_3 - outer face.

* Draw, if possible, two different planar graphs with same number of vertices, edges and faces.

* Draw, if possible, two different planar graphs with the same number of vertices and edges, but a different number of faces.

There is a connection between the number of vertices (v), the number of edges (e) and the number of faces (f) in any connected planar graph.

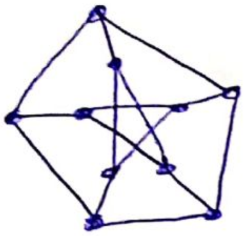
Euler formula:-

For any connected graph (planar graph) with v vertices, ~~and~~ e edges and f faces. we have $v - e + f = 2$.

Example:

verify whether the following are planar (or) not.

1. K_4 : Complete graph on 4 vertices.
2. C_5 : cycle on 5 vertices.
3. K_5 : Complete graph on 5 vertices.
4. Any tree.
5. Any Cycle.
6. $K_{2,2}$:
7. $K_{3,3}$:

8. 

2. Is it possible for a connected graph with 7 vertices and 10 edges to be drawn so that no edges cross and create 4 faces? Explain?

3. Prove that K_5 and $K_{3,3}$ are not planar?

4. Let G be a planar graph with $v \geq 3$ then $e \leq 3v-6$.

5a.

<u>Theorem</u> (Kuratowski's theorem)

[Necessary - sufficient condition for planar].

A graph is planar if and only if it does not contain any subdivision of K_5 (or) $K_{3,3}$.

1.2 Degree and handshaking

1.2.1 Definition of degree

Intuitively, the *degree* of a vertex is the “number of edges coming out of it”. If we think of a graph G as a picture, then to find the degree of a vertex $v\in V(G)$ we draw a very small circle around v, the number of times the G intersects that circle is the degree of v. Formally, we have:

Definition 1.2.1.

Let G be a simple graph, and let $v\in V(G)$ be a vertex of G. Then the ***degree of*** v, written $d(v)$, is the number of edges $e\in E(G)$ with $v\in e$. Alternatively, $d(v)$ is the number of vertices v is adjacent to.

Example 1

Note that in the definition we require G to be a simple graph. The notion of degree has a few pitfalls to be careful of G has loops or multiple edges. We still want to the degree $d(v)$ to match the intuitive notion of the “number of edges coming out of v” captured in the drawing with a small circle. The trap to beware is that this notion no longer agrees with “the number of vertices adjacent to v” or the “the number of edges incident to v”

Example 2.

Graph Isomorphisms

Generally speaking in mathematics, we say that two objects are "isomorphic" if they are "the same" in terms of whatever structure we happen to be studying. The symmetric group S_3 and the symmetry group of an equilateral triangle D_6 are isomorphic. In this section we briefly briefly discuss isomorphisms of graphs.

1.3.1 Isomorphic graphs

The "same" graph can be drawn in the plane in multiple different ways. For instance, the two graphs below are each the "cube graph", with vertices the 8 corners of a cube, and an edge between two vertices if they're connected by an edge of the cube:

Definition 1.3.3.

An isomorphism $\varphi:G\to H\varphi:G\to H$ of simple graphs is a biject $\varphi:V(G)\to V(H)\varphi:V(G)\to V(H)$ between their vertex sets that preserves the number of edges between vertices. In other words, $\varphi(v)\varphi(v)$ and $\varphi(w)\varphi(w)$ are adjacent in HH if and only if vv and ww are adjancent in $G.G.$

Example 1.

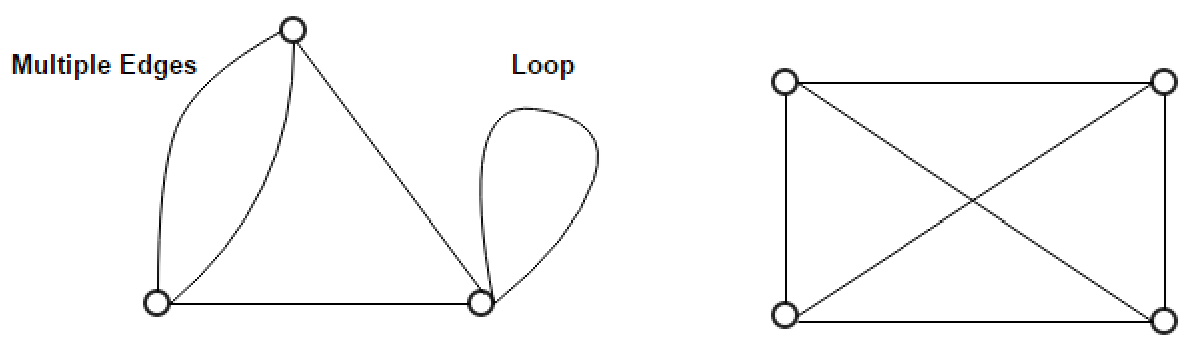
Although solving the graph isomorphism problem for general graphs is quite difficult, doing it for small graphs by hand is not too bad and is something you must be able to do for the exam. If the two graphs are actually isomorphic, then you should show this by exhibiting an isomrophism; that is, writing down an explicit bijection between their vertex sets with the desired properties. The most attractive way of doing this, for humans, is to label the vertices of both copies with the same letter set.

If two graphs are not isomorphic, then you have to be able to prove that they aren't. Of course, one can do this by exhaustively describing the possibilities, but usually it's easier to do this by giving an obstruction – something that is different between the two graphs. One easy example is that isomorphic graphs have to have the same number of edges and vertices. We'll discuss some others in the next section

Simple Graph

A **simple graph** is the undirected graph with **no parallel edges** and **no loops**.

A simple graph which has n vertices, the degree of every vertex is at most n -1.



Not a Simple Graph   Simple Graph

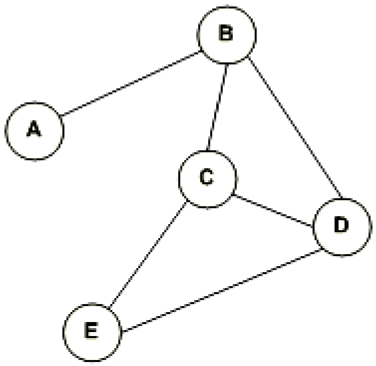
Example

In the above example, First graph is not a simple graph because it has two edges between the vertices A and B and it also has a loop. Second graph is a simple graph because it does not contain any loop and parallel edges.

Undirected Graph

An **undirected graph** is a graph whose edges are **not directed**.

Example



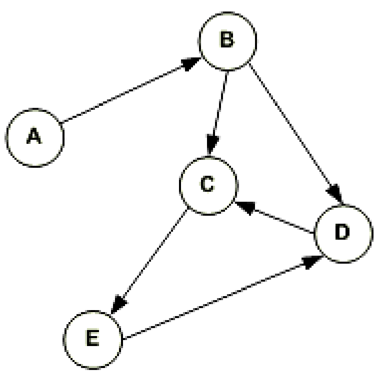
In the above graph since there is no directed edges, therefore it is an undirected graph.

Directed Graph

A **directed graph** is a graph in which the **edges are directed** by arrows.

Directed graph is also known as **digraphs**.

Example



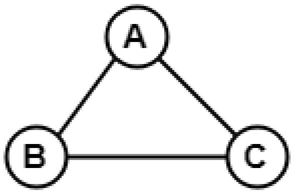
In the above graph, each edge is directed by the arrow. A directed edge has an arrow from A to B, means A is related to B, but B is not related to A.

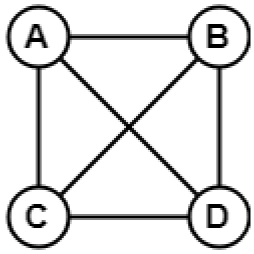
Complete Graph

A graph in which every pair of vertices is joined by exactly one edge is called **complete graph**. It contains all possible edges.

A complete graph with n vertices contains exactly nC2 edges and is represented by Kn.

Example



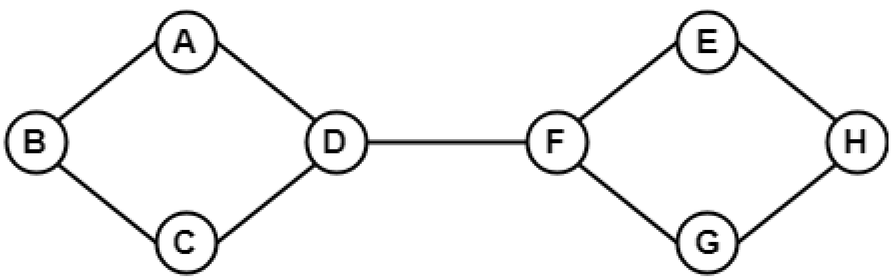


In the above example, since each vertex in the graph is connected with all the remaining vertices through exactly one edge therefore, both graphs are complete graph.

Connected Graph

A **connected graph** is a graph in which we can visit from any one vertex to any other vertex. In a connected graph, at least one edge or path exists between every pair of vertices.

Example

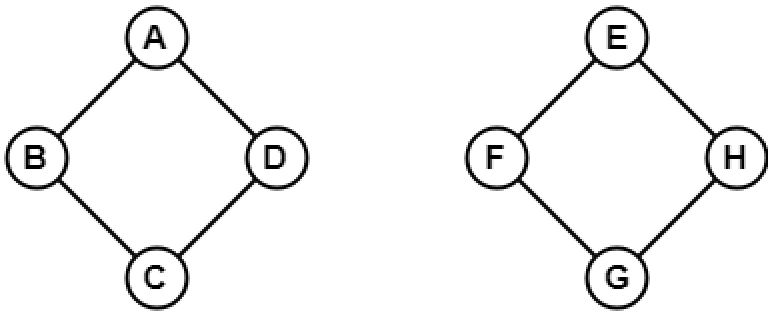


In the above example, we can traverse from any one vertex to any other vertex. It means there exists at least one path between every pair of vertices therefore, it a connected graph.

Disconnected Graph

A **disconnected graph** is a graph in which any path does not exist between every pair of vertices.

Example



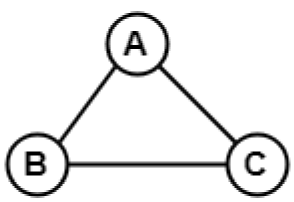
The above graph consists of two independent components which are disconnected. Since it is not possible to visit from the vertices of one component to the vertices of other components therefore, it is a disconnected graph.

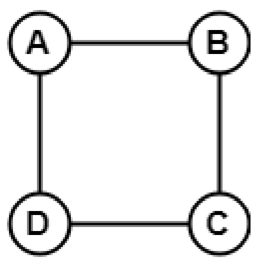
Regular Graph

A **Regular graph** is a graph in which degree of all the vertices is same.

If the degree of all the vertices is k, then it is called k-regular graph.

Example





In the above example, all the vertices have degree 2. Therefore they are called 2- **Regular graph**.

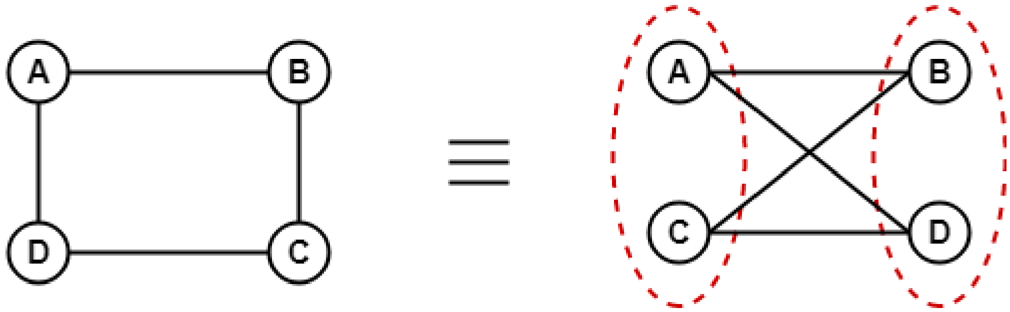
Bipartite Graph

A **bipartite graph** is a graph in which the vertex set can be partitioned into two sets such that edges only go between sets, not within them.

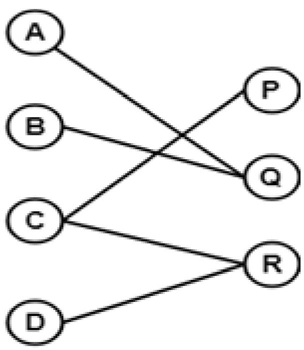
A graph G (V, E) is called bipartite graph if its vertex-set V(G) can be decomposed into two non-empty disjoint subsets V1(G) and V2(G) in such a way that each edge e ∈ E(G) has its one last joint in V1(G) and other last point in V2(G).

The partition V = V1 ∪ V2 is known as bipartition of G.

Example 1



Example 2



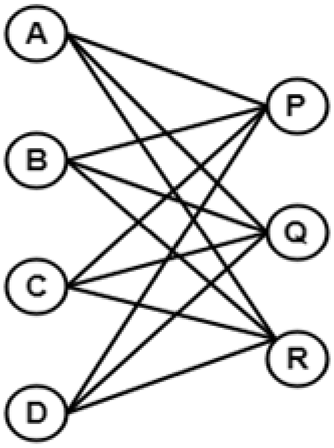
Complete Bipartite Graph

A **complete bipartite graph** is a bipartite graph in which each vertex in the first set is joined to each vertex in the second set by exactly one edge.

A complete bipartite graph is a bipartite graph which is complete.

1. Complete Bipartite graph = Bipartite graph + Complete graph

Example

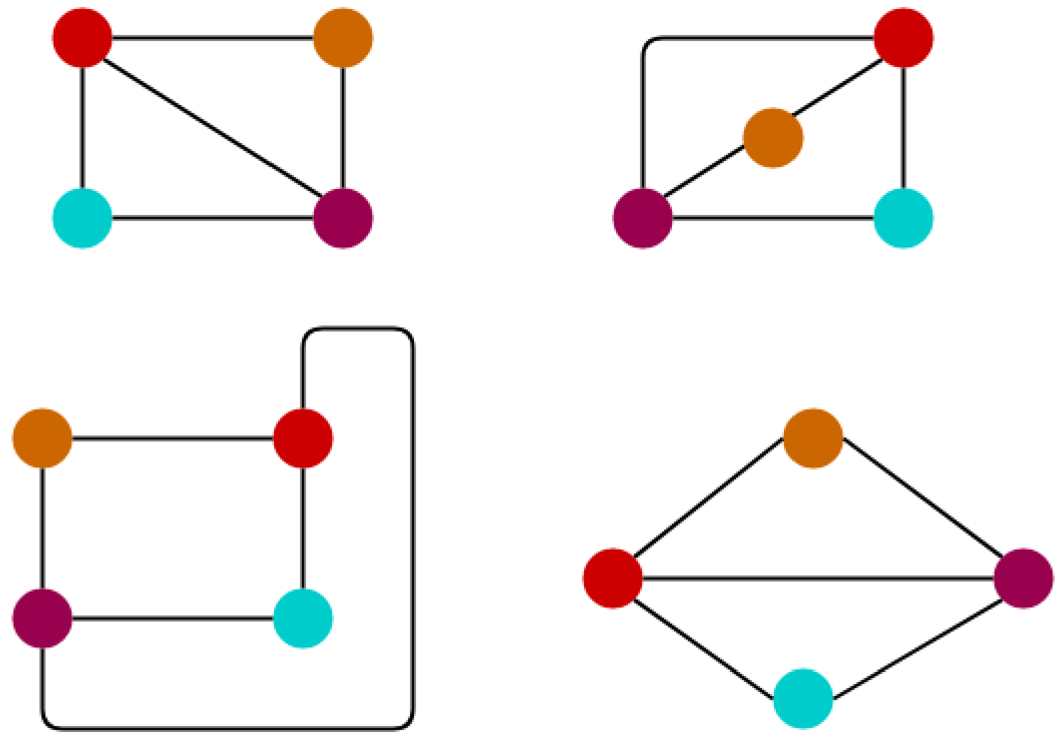


The above graph is known as $K_{4,3}$.

Graph Isomorphism-

Graph Isomorphism is a phenomenon of existing the same graph in more than one forms. Such graphs are called as **Isomorphic graphs**.

Graph Isomorphism Example-



Graph Isomorphism Example

Graph Isomorphism Conditions-

For any two graphs to be isomorphic, following 4 conditions must be satisfied-

- Number of vertices in both the graphs must be same.
- Number of edges in both the graphs must be same.
- Degree sequence of both the graphs must be same.
- If a cycle of length k is formed by the vertices { v_1 , v_2 , ….. , v_k } in one graph, then a cycle of same length k must be formed by the vertices { $f(v_1)$, $f(v_2)$, ….. , $f(v_k)$ } in the other graph as well.

Problem-01:

Are the following two graphs isomorphic?

Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 4
- Number of vertices in graph G2 = 4

Here,

- Both the graphs G1 and G2 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

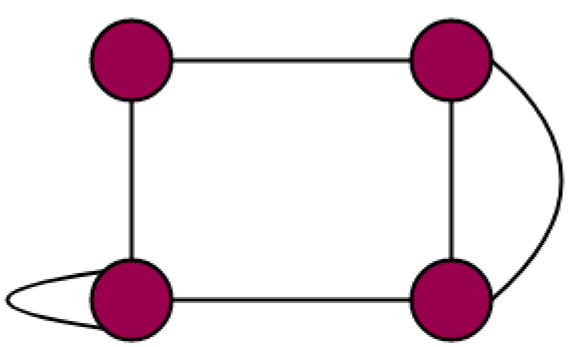
- Number of edges in graph G1 = 5
- Number of edges in graph G2 = 6

Here,

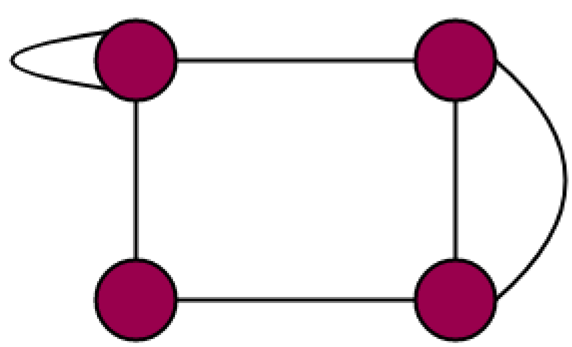
- Both the graphs G1 and G2 have different number of edges.
- So, Condition-02 violates.

Since Condition-02 violates, so given graphs can not be isomorphic.

∴ G1 and G2 are not isomorphic graphs.



G1



G2

Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 4
- Number of vertices in graph G2 = 4

Here,

- Both the graphs G1 and G2 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

- Number of edges in graph G1 = 5
- Number of edges in graph G2 = 6

Here,

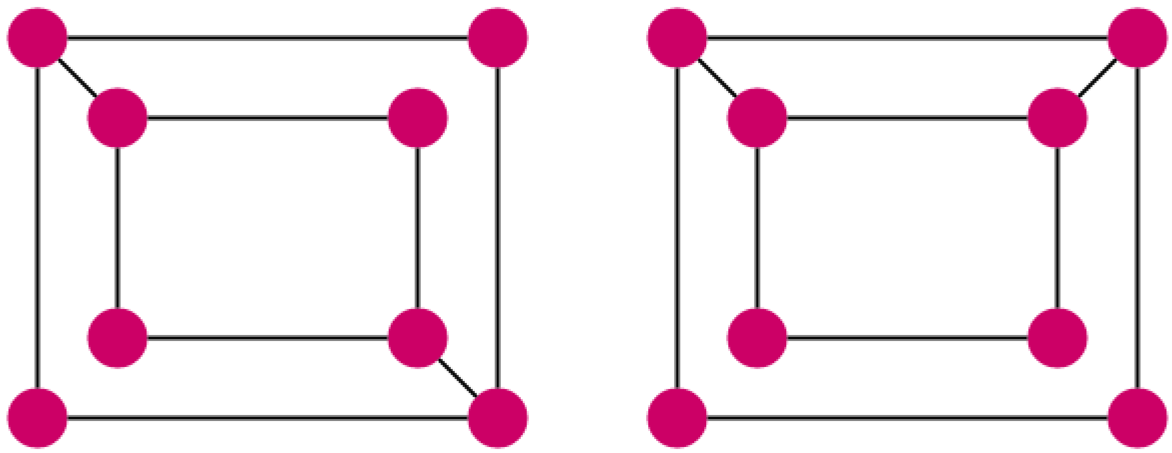
- Both the graphs G1 and G2 have different number of edges.
- So, Condition-02 violates.

Since Condition-02 violates, so given graphs can not be isomorphic.

∴ G1 and G2 are not isomorphic graphs.

Problem-02

Are the following two graphs isomorphic?



G1 G2

Solution-

Checking Necessary Conditions-

Condition-01:

- Number of vertices in graph G1 = 8
- Number of vertices in graph G2 = 8

Here,

- Both the graphs G1 and G2 have same number of vertices.
- So, Condition-01 satisfies.

Condition-02:

- Number of edges in graph G1 = 10
- Number of edges in graph G2 = 10

Here,

- Both the graphs G1 and G2 have same number of edges.
- So, Condition-02 satisfies.

Condition-03:

- Degree Sequence of graph G1 = { 2 , 2 , 2 , 2 , 3 , 3 , 3 , 3 }
- Degree Sequence of graph G2 = { 2 , 2 , 2 , 2 , 3 , 3 , 3 , 3 }

Here,

- Both the graphs G1 and G2 have same degree sequence.
- So, Condition-03 satisfies.

Condition-04:

- In graph G1, degree-3 vertices form a cycle of length 4.
- In graph G2, degree-3 vertices do not form a 4-cycle as the vertices are not adjacent.

Here,

- Both the graphs G1 and G2 do not contain same cycles in them.
- So, Condition-04 violates.

Since Condition-04 violates, so given graphs can not be isomorphic.

Kruskal’s Algorithm-

1. Kruskal’s Algorithm is a famous greedy algorithm.
2. It is used for finding the Minimum Spanning Tree (MST) of a given graph.
3. To apply Kruskal’s algorithm, the given graph must be weighted, connected and undirected.

Kruskal’s Algorithm Implementation-

Step-1:

Sort all the edges from low weight to high weight.

Step-2:

Take the edge with the lowest weight and use it to connect the vertices of graph.

If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.

Step-03:

Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

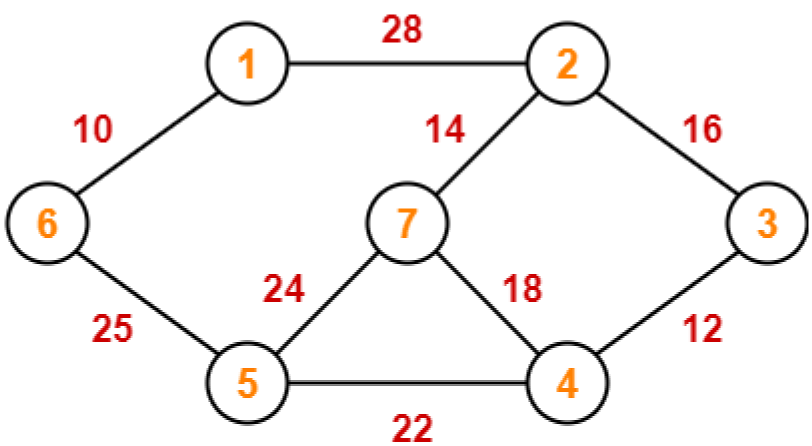
Special Case-

If the edges are already sorted, then there is no need to construct min heap.

So, deletion from min heap time is saved.
In this case, time complexity of Kruskal's Algorithm = O(E + V)

Problem-01:

Construct the minimum spanning tree (MST) for the given graph using Kruskal's Algorithm-

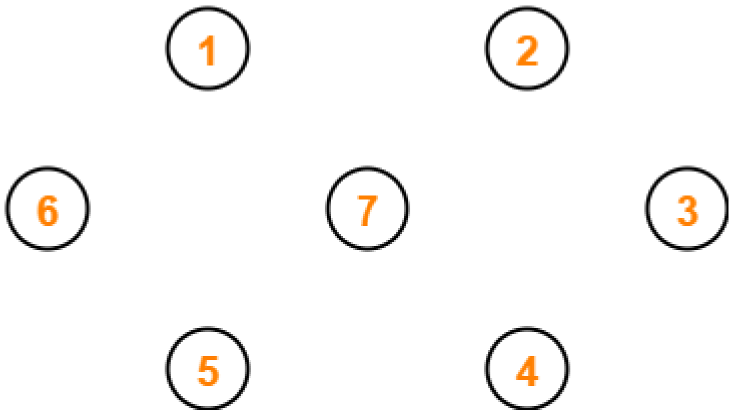


Solution-

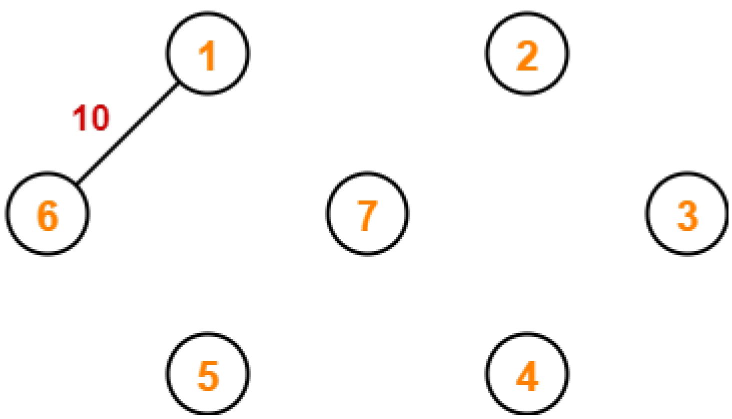
To construct MST using Kruskal's Algorithm,

1. Simply draw all the vertices on the paper.
2. Connect these vertices using edges with minimum weights such that no cycle gets formed.

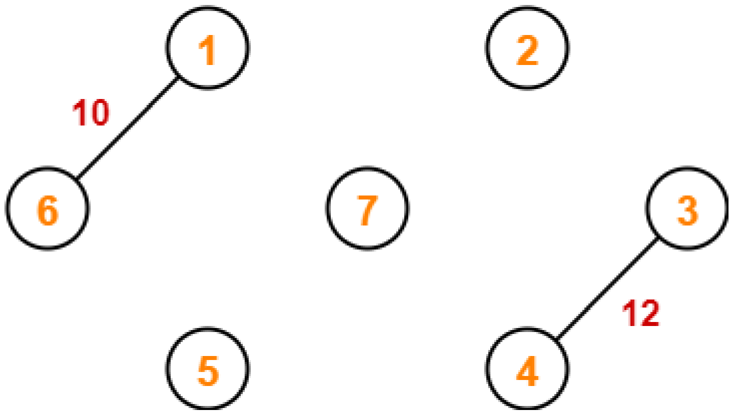
Step-01:



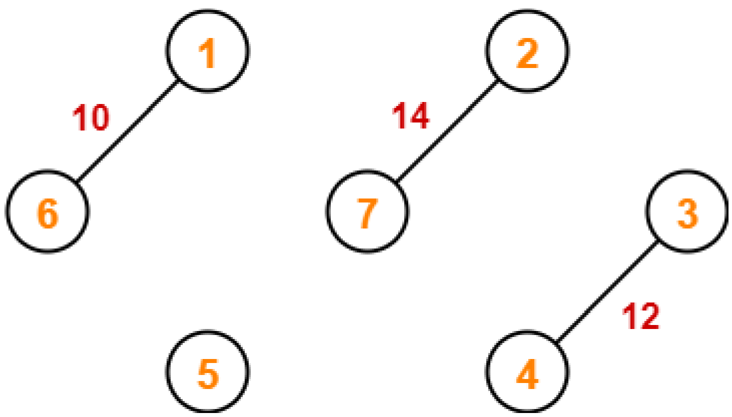
Step-02:



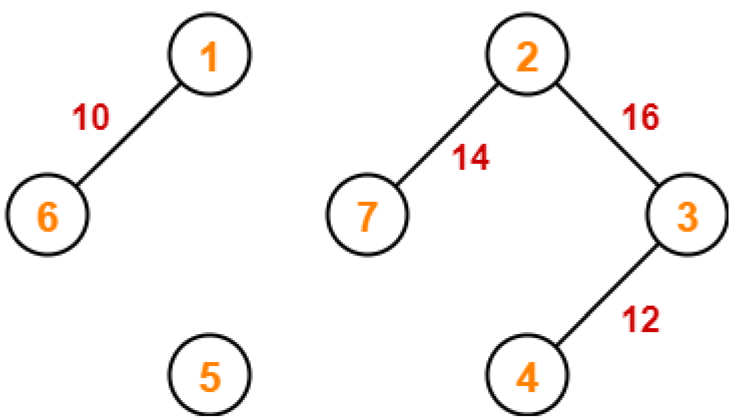
Step-03:



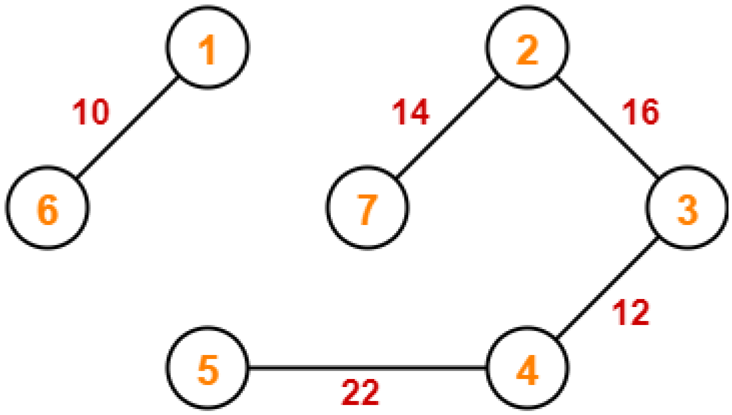
Step-04:



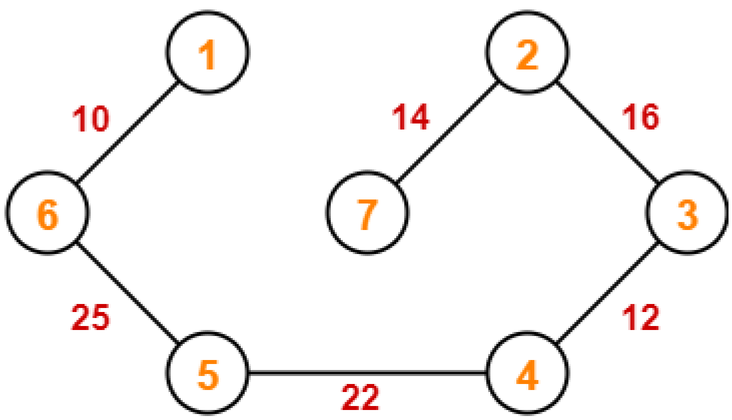
Step-05:



Step-06:



Step-07:



Since all the vertices have been connected / included in the MST, so we stop.

Weight of the MST

= Sum of all edge weights

= 10 + 25 + 22 + 12 + 16 + 14

= 99 units

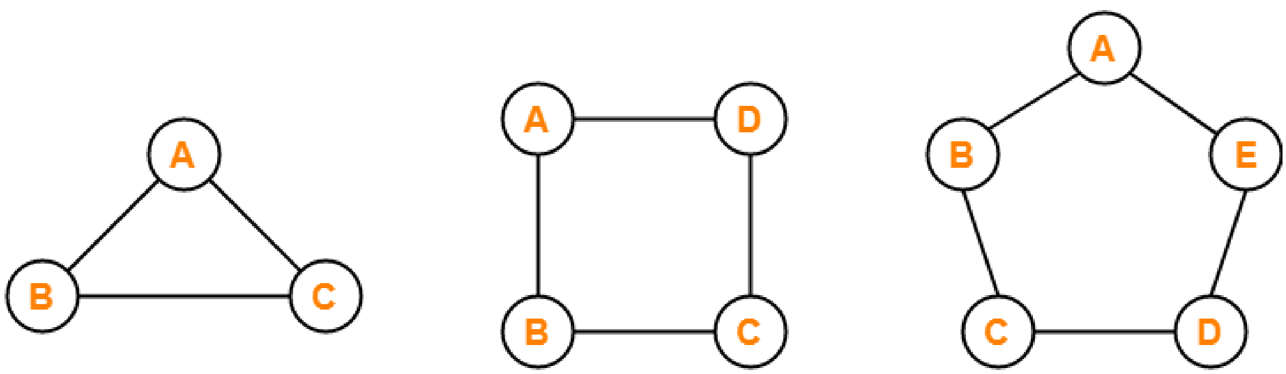
Chromatic Number-

- Graph Coloring is a process of assigning colors to the vertices of a graph.
- It ensures that no two adjacent vertices of the graph are colored with the same color.
- Chromatic Number is the minimum number of colors required to properly color any graph.

Cycle Graph-

- A simple graph of ‘n’ vertices (n>=3) and n edges forming a cycle of length ‘n’ is called as a cycle graph.
- In a cycle graph, all the vertices are of degree 2.

Examples-



Examples of Cycle Graph

In these graphs,

- Each vertex is having degree 2.
- Therefore, they are cycle graphs.

Path

In graph theory, a path is defined as an open walk in which-

- Neither vertices (except possibly the starting and ending vertices) are allowed to repeat.
- Nor edges are allowed to repeat.

Circuit

In graph theory, a circuit is defined as a closed walk in which-

- Vertices may repeat.
- But edges are not allowed to repeat.

Four-Color Theorem. The four-color theorem states that any map in a plane can be colored using four-colors in such a way that regions sharing a common boundary (other than a single point) do not share the same color.