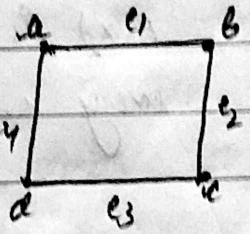
22

DATE: / /

Graph Theory

V → set of vertices (non-empty)
E → set of edges
↑ paired element of V.

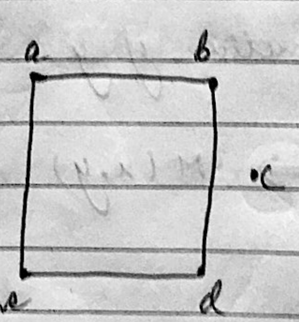
G(V,E)
Graph, which is collection of vertices & edges



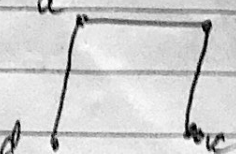
$G(V,E)$
$G(V) = \{a, b, c, d\}$
$G(E) = \{e_1, e_2, e_3, e_4\}$

02-22

★ **Isolated vertex** –
is a vertex which has no connected edge. (degree is zero)



2) **Pendent vertex** –
Vertex having degree 1 or which has only one edge connected to it.



Here c, d are pendent vertices.

DATE: / /

3) <u>Null graph</u> -

The graph having no edge, is null graph. or the graph having only vertices and whose edge set is empty.

<u>Eg</u>

• a

c • • b

→ Types

a) Single vertex null graph (having single vertex)

• a

b) Double vertex null graph (having double vertex)

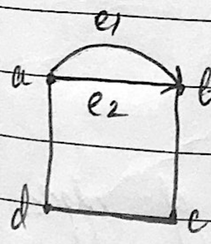
• a • b

4) <u>Non-Empty Set</u> — Set having atleast one element is a non-empty set.

5) <u>Empty set</u> — Set having no element. It is denoted by 'ϕ'

$A = \{ \phi \}$

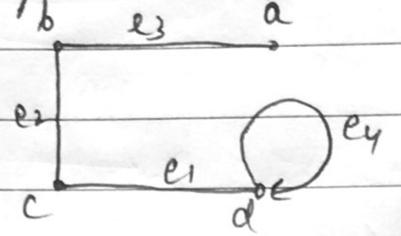
6) <u>Parallel edge</u> - When more than one edges exists b/w 2 adjacent vertices, then those edges are said to be parallel edges.



Here, e_1, e_2 are parallel edges.

7) <u>Loops -</u>

When an edge have same starting and ending vertex, then it is known as loop.

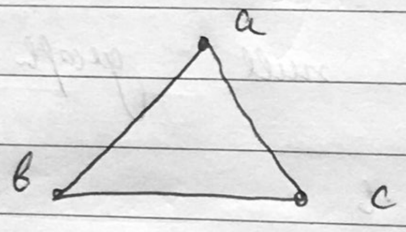


Here, e4 forms a loop.

8) <u>Simple Graph -</u>

Graph having no self loops and no parallel edges, is called a simple graph.

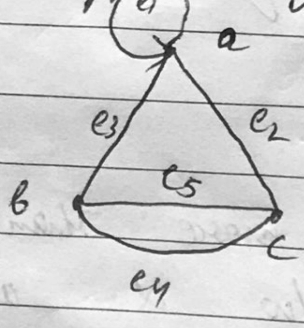
<u>E.g</u>



9) <u>Multi Graph -</u>

A graph having self loops and parallel edges is called a multigraph.

It is opposite of simple graph.

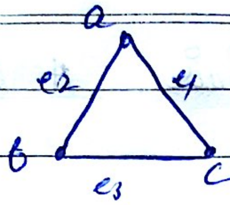


10) <u>Finite Graph -</u>

The graph having finite number of edges and vertices.

DATE: / /

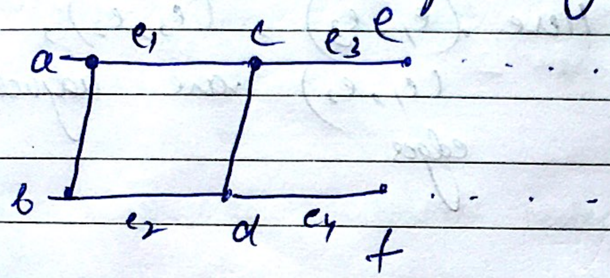
e.g.



$G(V,E) = G(V) = \{a, b, c\}$
$G(E) = \{e_1, e_2, e_3\}$

11) <u>Infinite graph</u> -

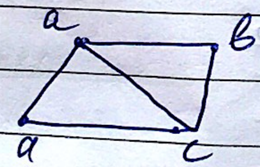
Graph having infinite number of vertices and edges is called an infinite graph.



$G(V) = \{a, b, c, d, e, f \dots\}$
$G(E) = \{e_1, e_2, e_3, e_4, \dots\}$

12) <u>Order of Graph</u> -

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



$G(V,E)$
order = 4
size = 5

13) <u>Size of the graph</u> -

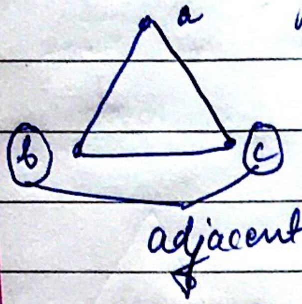
The number of edges in a graph is called the size of the graph.

In the above graph, the size is 5.

14) <u>Adjacent vertex</u> -

Vertices connected by the same edge are called adjacent vertices.

They always exist in pair. A loop is not considered as adjacent vertex.

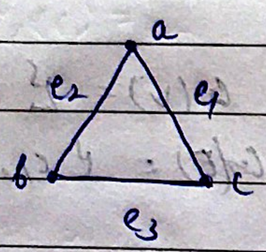


G (V, E)

(a,b) (a,c) (b,c)

5) <u>Adjacent Edges</u> –

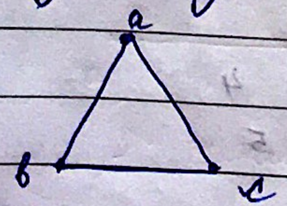
Two non-parallel edges are said to be adjacent, if they are connected to a same vertex.



Here (e_1, e_3), (e_3, e_2), (e_1, e_2) are adjacent ~~vert~~ edges.

15) <u>Degree of a vertex</u> –

The no. of edges coming towards a vertex or connected to that vertex, is the degree of that vertex.

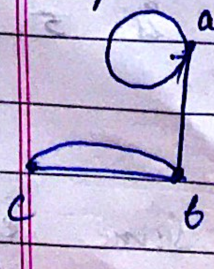


deg(a) = 2

deg (b) = 2

deg (c) = 2

A loop constitutes '2' to the degree and a parallel edge constitutes ~~to~~ '1'.



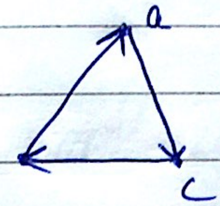
deg (a) = 3 (containing parallel edge)

deg (c) = 2

deg (b) = 3

17) <u>Directed Graph</u>

A graph having direction of its edges or whose edges have direction is a directed graph.



Here, $G(V,E)$ is a directed graph.

18) <u>Initial Vertex</u> -

Vertex from which edge starts is known as initial vertex

$a \bullet \longrightarrow b$

Here, a is initial vertex and b is terminal vertex.

19) <u>Terminal Vertex</u> -

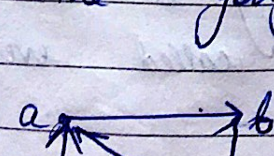
Vertex to which an edge ends is called terminal vertex.

20) <u>Indegree</u> -

The <u>coming edge</u> constitutes to the indegree

21) <u>Outdegree</u> -

The going edge constitutes to the outdegree.



$\deg(a) = $ out $= 1$, in(deg) $= 2$

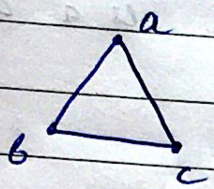
$\deg^-(a) = 2$

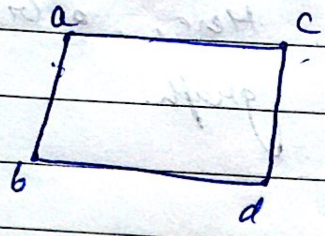
$\deg^+(a) = 1$

A loop constitutes, 1 to the indegree $(-)$ and 1 to the outdegree $(+)$. In total $= 2$

22) **Regular Graph:-**

$G(V, E) \longrightarrow$ should be a simple graph. and for a regular graph, the degree of all the vertices should be same.





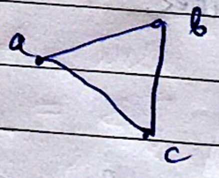
deg(a) = 2
deg(b) = 2
deg(c) = 2

deg(a) = 2
deg(b) = 2
deg(c) = 2
deg(d) = 2

23) **r-regular graph:-**

A regular graph depicting the degree of vertex of ~~that~~ graph is called r-regular graph. Here, 'r' represents the degree.

1- regular graph



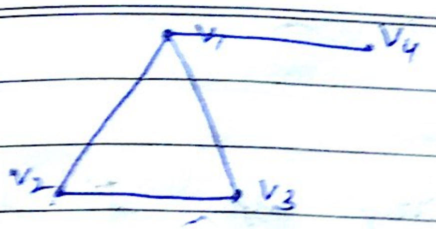
2-regular graph

24) **Even vertex**

In a graph, if the degree of a vertex is an even integer then, it is called even vertex.

25) **Odd Vertex**

If in a graph, the degree of a vertex is an odd integer then it is called odd vertex.



deg (v1) = 3
deg (v2) = 2 ⎤ even vertex
deg (v3) = 2 ⎦
deg (v4) = 1

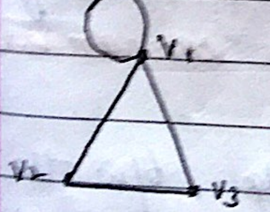
odd vertex.

Hand- Shaking Theorem-

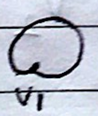
Theorem- The sum of the degree of all the vertices in a graph is equal to twice the no: of edges in a graph.

Proof- Let G(V, E) be a graph, where
$V = \{v_1, v_2, v_3, \dots, v_n\}$ with set of vertices
$E = \{e_1, e_2, e_3, \dots, e_n\}$ with set of edges.

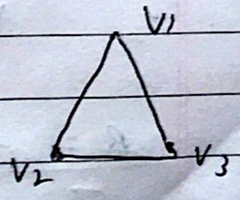
We know that, every edge ~~line~~ lie b/w two vertex, so it provide degree 1 to each vertex. Hence every edge contribute degree two for the graph. So, sum of degree of all the vertices is equal to ~~the~~ twice the no. of edges in a graph.



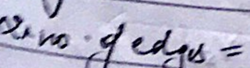
deg(v1) = 4
deg(v2) = 2
deg(v3) = 2
deg(V) = 4+2+2 = 8
2 × no. of edges = 4×2 = 8



deg(v1) = 2
edge = 1
2× edge = 2×1



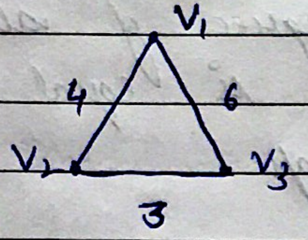
deg(v1) = 2
deg(v2) = 2
deg(v3) = 2
deg(V) = 6
edge = 3
2× edges = 6
same



deg(v1) = 1 deg(v2) = 1 deg(V) = 2 2×edge = 2×1 = 2

☆ Weighted graph -

In a graph, if every edge is assigned with ~~an~~ a positive integer no., then the graph is called weighted graph.



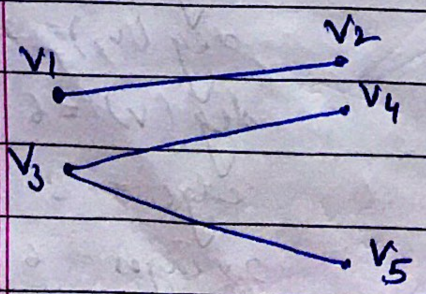
Weight of $(V_1, V_2) = 4$
$(V_1, V_3) = 6$
$(V_2, V_3) = 3$

☆ Bipartite graph -

A simple graph $G(V, E)$ is said to be bipartite if the set of vertices (V) can be divided into 2 non empty subsets V_1 & V_2 such that

A = every vertex of V_1 is connected with some vertex of V_2 or ~~some~~ every vertex of V_2 is connected with some vertex of V_1. $[V_1 \cap V_2 = \phi]$ // no common vertex

B = there should be no edge b/w the vertices of set V_1 (itself) and set (V_2) itself.



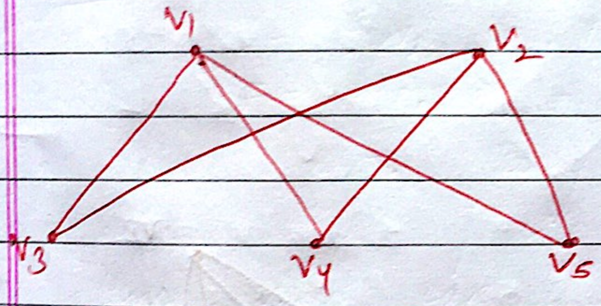
$V = \{V_1, V_2, V_3, V_4, V_5\}$
$V_1' = \{V_1, V_3\}$
$V_2' = \{V_2, V_4, V_5\}$

PAGE NO.:
DATE: / /

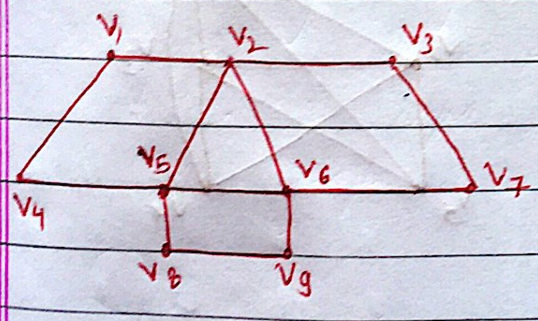
* Complete Bipartite graph → (V_1, V_2, E)

A graph $G(V, E)$ is called a complete bipartite graph, if vertices V can be partitioned into two subsets V_1 & V_2 such that each vertex V_1 is connected to each vertex of V_2.

It is denoted by 'K_{mn}' where 'm' is the no. of vertices of set V_1 while 'n' denotes the no. of vertices of set V_2.



K_{23}
$V_1' = \{ V_1, V_2 \}$
$V_2' = \{ V_3, V_4, V_5 \}$



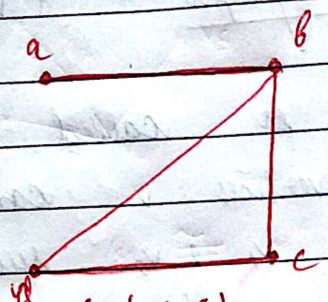
$V_1 = \{ V_1, V_3, V_5, V_9 \}$
$V_2 = \{ V_2, V_4, V_8, V_6, V_7 \}$

* Complementary Graph - // (complement of a graph)

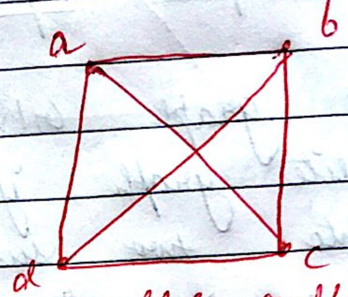
Let $G(V,E)$ be a simple graph with m- vertices & e-edges, then the complementary graph $G(V, E)$ is a graph which contain all the vertices of G & all those edges which are not in G but exist in K_n (complete graph)

∴ will contain edges, other than the edges present in G. like {A ∸ A}

E.g

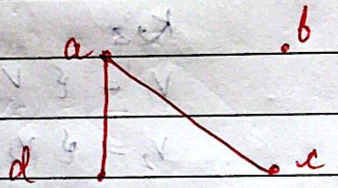


$G(V,E)$

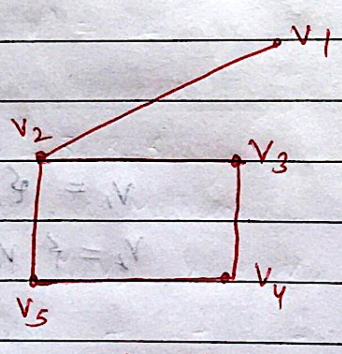


complete graph
$K_4 = (G(V,E))$

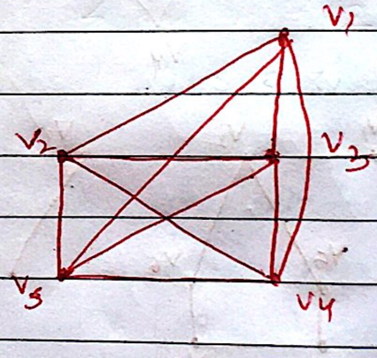
complement of a graph = complete graph − given graph
$= K_4 - G(V,E)$

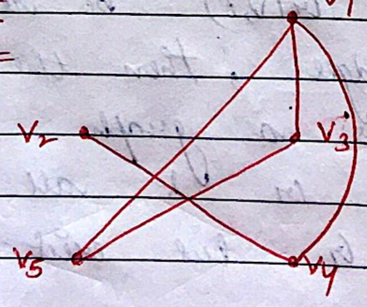
=


e.g



$G(V,E)$



Complement
$\overline{G}(V,E) =$


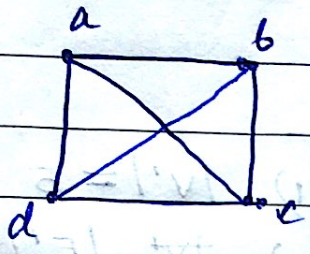
Note

1) Let $G(V,E)$ be a graph of e-edges & n-vertices then no of edges in a complementary graph $\overline{G}$ is $\left[\frac{n(n-1)}{2} - e\right]$

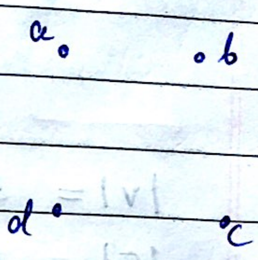
PAGE NO.:
DATE: / /

2) The complementary graph of complete graph i.e. K_n with 'n' vertices & $\frac{n(n-1)}{2}$ edges have only n-vertices & no edges.

K_4 =



$\overline{G_1}$ =



with 4 vertices and 6 edges

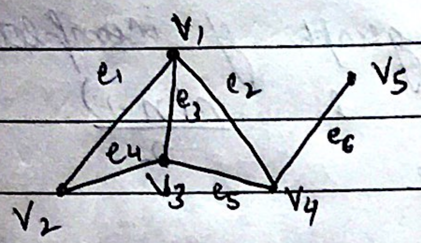
without any edge.

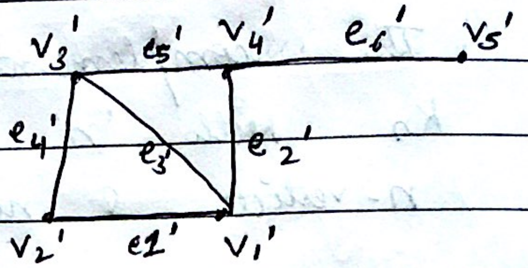
3) If in a graph, & 'x' be the degree of any vertex then degree of this vertex in complementary graph is $[n-x-1]$.

Isomorphic Graph –

- No. of vertex should be equal
- No. of edges should be equal.
- Sum of degree of the vertex should be equal.
- Degree sequence should be same
- Adjacency matrix should be same
- If G_1 contains some vertices with any degree 'x' then, it is must that G_2 contains a vertex with degree 'x', then ~~it~~ ~~is~~, i.e, same degree vertex should be present in both.
- Travelling should be equal (i.e, degree of walk should be same).

E.g –





core $G \cong G'$

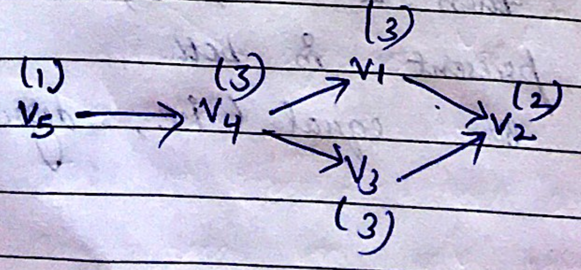
1) $|V| = 5$
2) $|E| = 6$
3) $\deg(v_1) = 3$
$\deg(v_2) = 2$
$\deg(v_3) = 3$
$\deg(v_4) = 3$
$\deg(v_5) = 1$

4) $\sum \deg(V) = 12$
5) degree sequence
$(1, 2, 3, 3, 3)$

6) Adjacency Matrix

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

7)



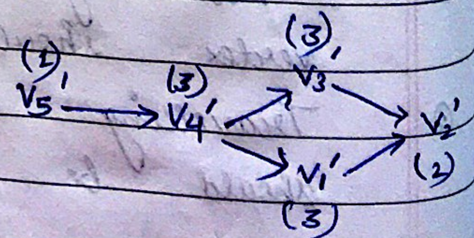
1) $|V'| = 5$
2) ~~$|V'|$~~ $|E'| = 6$
3) $\deg(v_1') = 3$
$\deg(v_2') = 2$
$\deg(v_3') = 3$
$\deg(v_4') = 3$
$\deg(v_5') = 1$

4) $\sum \deg(V') = 12$
5) $(1, 2, 3, 3, 3)$

6)

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

7)



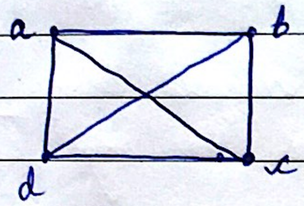
Hence G and G' are isomorphic graphs

☆ complete graph

A complete graph is a graph in which exactly one edge is present between every pair of vertices or every vertex is connected to another vertex with an edge.

A complete graph of having 'n' vertices is represented by 'K_n'
The degree of each vertex in a complete graph is $(n-1)$.

K_4



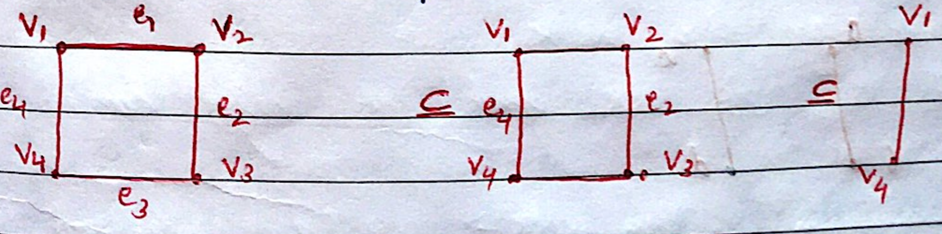
No. of edges $= \frac{n(n-1)}{2}$

$deg(a) = 3$ $deg(b) = 3$ $deg(c) = 3$ $deg(d) = 3$

☆ Sub graph -

Let $G(V, E)$ be a graph, then $S(V_1, E_1)$ is said to be subgraph of G if V_1 is subset of V and E_1 is subset of E, i.e $V_1 \subseteq V$ & $E_1 \subseteq E$

$V_1 \subseteq V \neq \phi$ $E_1 \subseteq E \neq \phi$



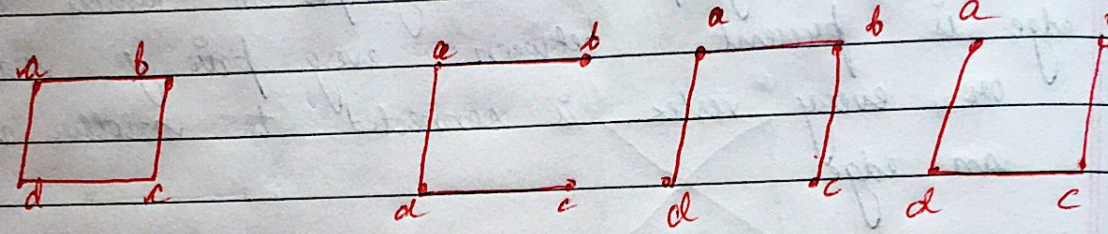
☆ Super Graph

If $G(V, E)$ be a graph & $S(V_1, E_1)$ be a subgraph of G, then G is the super graph of $S(V_1, E_1)$.

☆ Spanning sub Graph -

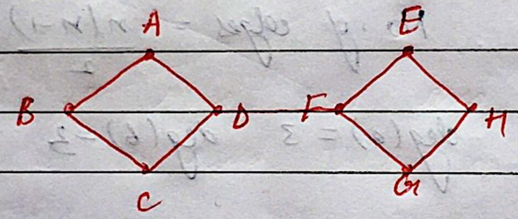
A subgraph $S(V_1, E_1)$ containing all vertex of graph G without circuit is called spanning subgraph of G.

// should contain every vertex but not a cycle



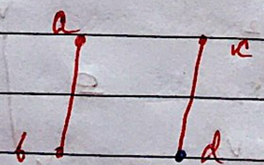
★ Connected Graph -

A graph G(V, E) is said to be connected graph if there is a path b/w every pair of vertex



★ Disconnected Graph -

A graph G(V, E) is said to be disconnected, if there doesn't exist a path b/w every pair of vertices



★ **Walk →** In a graph G(V, E), an alternative finite sequence of vertices & edges is called a walk.

- It is denoted by 'w' & it starts from vertex and ends on a vertex
- The starting vertex is called initial or origin vertex and the ending vertex is called

PAGE NO.:
DATE: / /

terminal vertex.

⊙ The no. of edges included in a walk is called length of the walk.

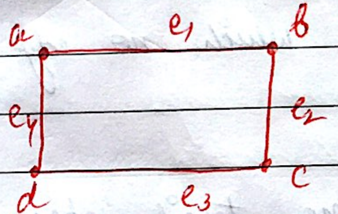
★ Open walk -

When initial and terminal vertex of a walk are not same, then it is open walk.

★ Closed Walk -

When initial and terminal vertex of a walk ~~is~~ are same, then it is called closed walk.

The walk starts and ends on same vertex.



open walk - $a e_1 b e_2 c e_3 d$

closed walk - $\underbrace{a e_1 b e_2 c e_3 d e_4 a}$

this doesn't indicates the ~~repeat~~ repetitiveness of a vertex 'a' but it indicates the closeness of a walk.

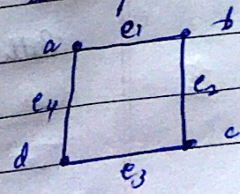
★ Trivial Walk -

When no edge exists in a graph and no edge exists in a walk. or

If the length of walk = 0, then it is trivial walk.

★ ~~Trial~~ Trail -

An open walk in which no edge is repeated, is termed as trail



$a e_1 b e_2 c e_3 d$ ~~e_4~~ // open walk + trail

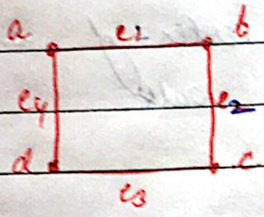
{vertex can be repeated}

as no edge is repeated.

★ Circuit

A closed trail is called a circuit

- closed walk
- without edge repetation
- vertex can be repeated



$W_1 \Rightarrow a e_1 b e_2 c e_3 d e_4 a$

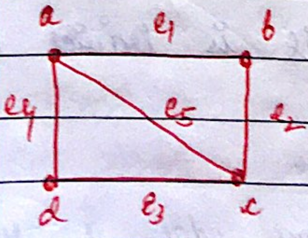
$W_2 \rightarrow a e_1 b e_2 c e_3 d e_4 a e_1 b$

W_1 is a closed walk and a circuit because it is a closed trail in which no edge is repeated

W_2 is an open walk → no trail because e_1 edge is repeated

★ Path

- It is an open walk
- In this <u>no edge</u> and <u>no vertex</u> can be repeated



open walk → $a e_1 b e_2 c e_5 a e_4 d$

closed walk → $a e_1 b e_2 c e_3 d e_4 a$

Trial → $a e_1 b e_2 c e_3 d$

circuit → $a e_1 b e_2 c e_3 d e_4 a$

Path → ~~$a e_1 b e_2 c e_5 a$~~ $a e_1 b e_2 c e_3 d$

Cycle -

~~An open~~ closed path (i.e) in which no edges and no vertices are repeated is called a cycle

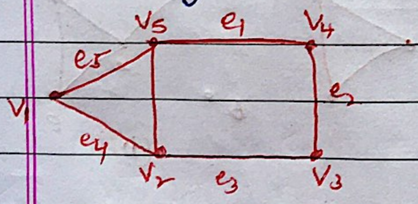
Note

A walk is always a trail F
A trail is always a walk T
A path is always a walk T
A walk is always a path X

Distance b/w Vertices -

The length of the shortest path b/w two vertex of connected graph, is called Distance and it is denoted by $d(u,v)$ where d is distance and u & v are vertices.

$u \rightarrow$ initial vertex
$v \rightarrow$ final vertex

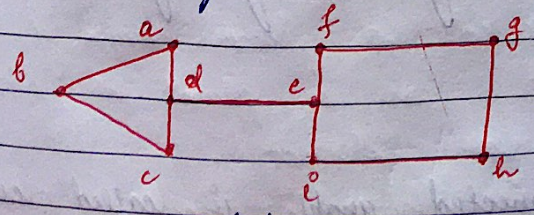


$d(v_1, v_5) = 1$
$d(v_1, v_2) = 1$
$d(v_1, v_4) = 2$
$d(v_1, v_3) = 2$

$d(v_2, v_3) = 1$
$d(v_2, v_5) = 1$
$d(v_5, v_4) = 1$

Bridge -

An edge is called bridge if removal of it converts a connected graph into a disconnected graph or in two components



$G(V, E)$

Here (ed) is acting as a bridge, as removing (ed) will ~~turn~~ turn graph $G(V,E)$ ~~into~~ in 2 graphs or a disconnected graph.

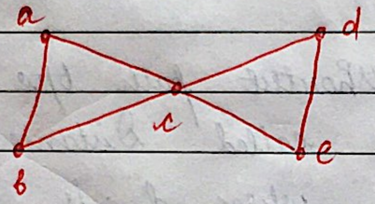
PAGE NO.:
DATE: / /

☆ Edge connectivity

Let G(V,E) be a connected graph and ~~there~~ the minimum number of edges whose removal makes 'G' disconnected is called edge connectivity of G.
represented by $\lambda(G)$

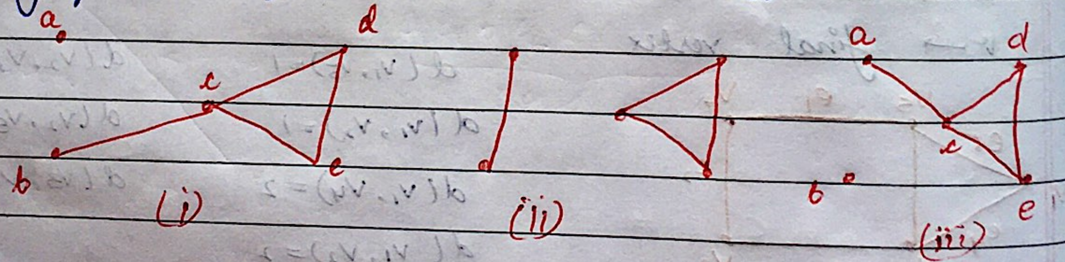
Or, the number of edges in a smallest cut set of G is called the edge connectivity of G.

E.g



G(V, E)

By removing two minimum edges, the connected graph G(V,E) becomes disconnected



☆ Cut set –

A set of edges of a graph is called cut set of 'G' if removal of set of edges from G converts it from connected to disconnected graph. But not any subset of set of edges can do this

☆ Cut Vertex

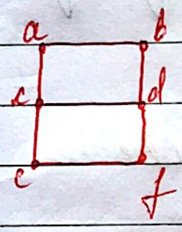
The vertex of connected graph is called cut vertex, if after deleting this vertex graph G.

is disconnected.

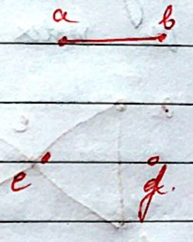
☆ **Vertex connectivity -**

The minimum no. of vertex after removal of which a connected graph G is converted into disconnected graph with two or more components, these is vertex connectivity.

The edges connected to that vertex got automatically removed.



vertex connectivity = 2

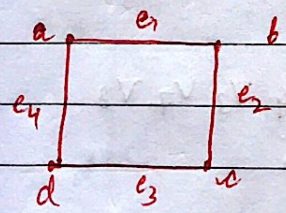


V = {c, d}.

☆ **Euler circuit -**

A circuit of connected graph is called euler circuit if it contains all the edges of G without any edge repeatation.

{ closed walk, no edge repeatation, contains all edges



E.C = $a e_1 b e_2 c e_3 d e_4 a$.

☆ **Euler graph -**

A graph in which a Euler circuit exists, is Euler graph.

☆ **Euler path -**

Open euler circuit is called euler path.

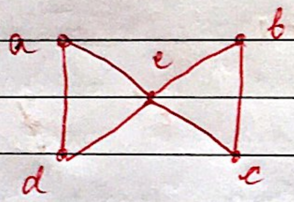
PAGE NO.:
DATE: / /

☆ Hamilton Graph:-

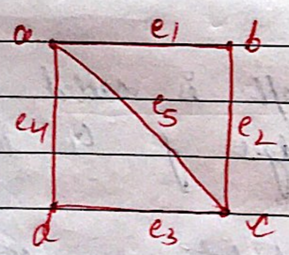
A graph is called hamilton graph if it contains hamiltons cycle
{ closed walk, no edge & no vertices is repeated, all ~~edges~~ vertices should be present

☆ Hamilton cycle:-

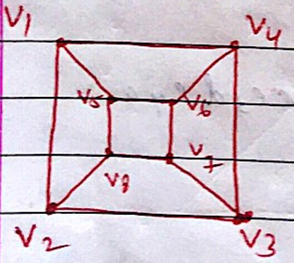
A cycle of connected graph is called hamilton cycle if, it contains all the vertex of G without any repeatation of edge or vertex



Euler graph – eadecbe



$\underbrace{a\,e_1\,b\,e_2\,c\,e_3\,d\,e_4\,a}_{\text{closed walk}}$ – cycle

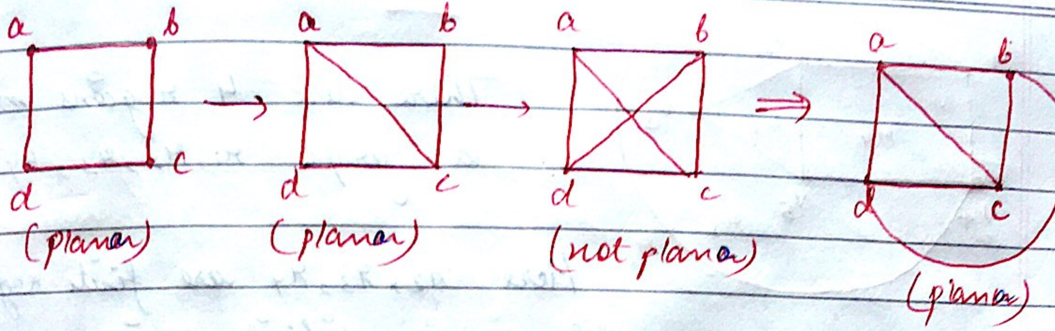


$V_8\,V_2\,V_1\,V_5\,V_6\,V_4\,V_3\,V_7\,V_8$
= Hamilton graph

☆ Planar Graph:

A graph is called a planar graph if it can be represent in a plane in which no two of its edges intersect

A graph is a planar graph if its edges do not intersect

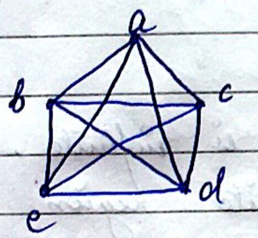


// edges doesn't intersect

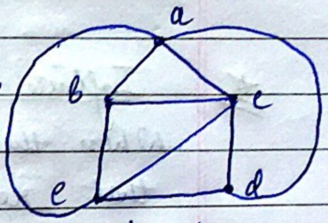
A graph is called planer graph if it can be represented in a plane

Q Is K_5 a planer graph or not

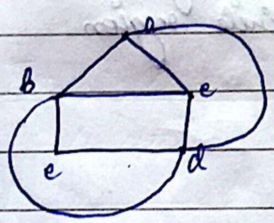
K_5 =

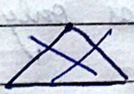


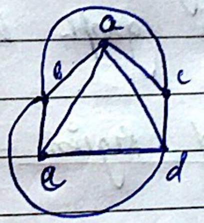
→ Non-planer graph



Non planer

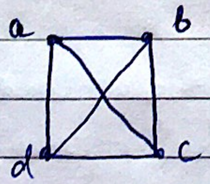


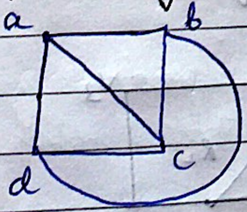




as it can't be transformed into a ~~non~~ planer graph.

Q Is K_4 is a planer graph or not





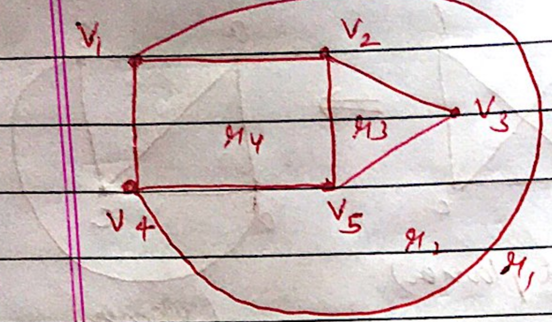
It is a planer graph

Regions

A planer graph ~~is~~ divides the planes into one or more regions.

A regions can be defined as an area of the plane that is bounded by edges and cannot be further subdivided.

(Regions are always for planer graph & not for non-planer)



There are 4 regions in a graph r_1, r_2, r_3, r_4

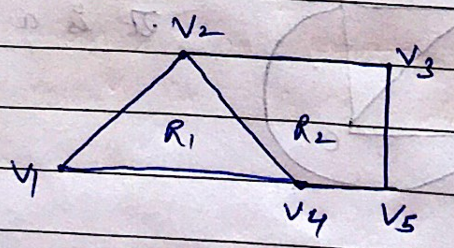
Here r_2, r_3, r_4 are finite regions while r_1 is infinite region

☆ **finite region** -
When the area of the region is finite, then it is called finite region

☆ **Infinite region**
When the area of the region is infinite, then that region is called infinite region
A planar graph has only one infinite region

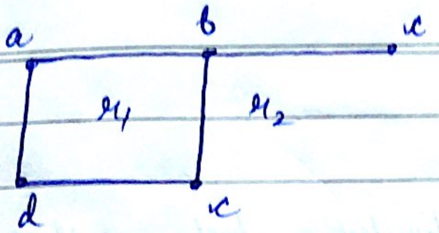
☆ **Degree of Region** -

Degree of region is equal to no. of edges enclosing that region.
for e.g.



$\left.\begin{array}{l} \deg(r_1) = 3 \\ \deg(r_2) = 4 \end{array}\right\}$ degree of bounded region

\# Degree of unbounded region = degree of (r) or no. of edges enclosing the region r.

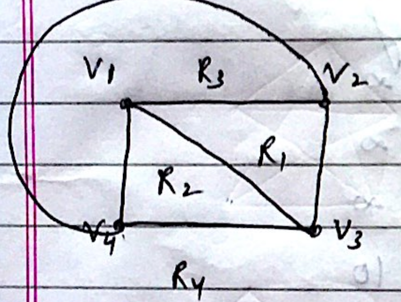


$deg(r_1) = 4$
$deg(r_2) = 6$

Euler's formula –

Let G be a connected planar simple graph with 'e' edges and 'v' vertices. Let 'r' be the no. of regions in planar representation of G. Then,

$\boxed{r = e - v + 2}$ or $\boxed{v - e + r = 2}$



$e = 6$
$v = 4$
$r = 4$

$r = -v + e + 2$ or $\boxed{r = e - v + 2}$

$\Rightarrow r = 6 - 4 + 2$

$\boxed{r = 4}$ // direct formula to calculate regions

Q If there are 20 vertices, each of degree 3, then in how many regions does a representation of this planar graph splits the plane.

We know, $v = 20$

$\boxed{\Sigma deg(v) = 2e}$ // By Handshaking theorem

$\Sigma deg(v) = 20 \times 3 = 60$ (Sum of degree)

$\therefore e = 60/3 = 30$, $e = 30$

$\boxed{r = e - v + 2}$ // by Euler's formula

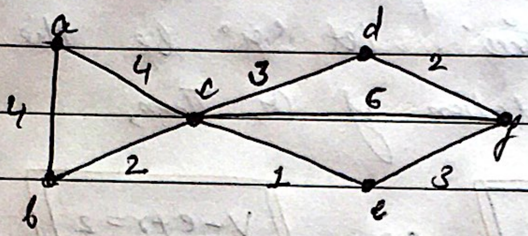
$r = 30 - 20 + 2$

$\boxed{r = 12}$

Hence, 12 regions will be formed

★ Dijkstra's Algorithm

It is also known as shortest path algorithm which is used to find the shortest path between any two vertices of a graph



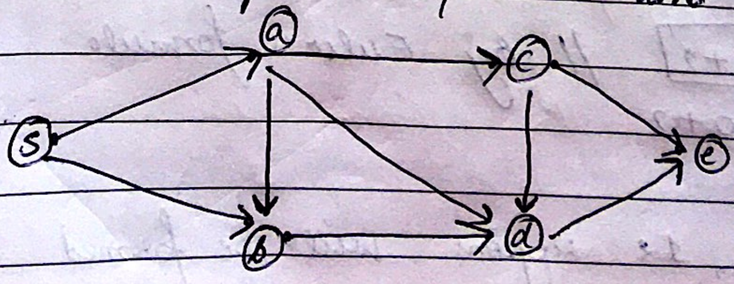
a	b	c	d	e	f
$\boxed{0}$ ✓	∞	∞	∞	∞	∞
$\boxed{0}$	$\boxed{4}$ ✓	4	∞	∞	∞
$\boxed{0}$	$\boxed{4}$	$\boxed{4}$ ✓	∞	∞	∞
$\boxed{0}$	$\boxed{4}$	$\boxed{4}$	7	$\boxed{5}$ ✓	10
$\boxed{0}$	$\boxed{4}$	$\boxed{4}$	$\boxed{7}$ ✓	$\boxed{5}$	8
$\boxed{0}$	$\boxed{4}$	$\boxed{4}$	$\boxed{7}$	$\boxed{5}$	$\boxed{8}$ ✓

Shortest distance – 8 units
Shortest path – $a \to c \to e \to f$

Note

It is only for ~~und~~ weighted graph and can be used for both directed and undirected graph

Q. Find shortest path b/w s and e.



s	a	b	c	d	e
$\boxed{0}$	∞	∞	∞	∞	∞
$\boxed{0}$	$\boxed{1}$	5	∞	∞	∞
$\boxed{0}$	$\boxed{1}$	3	3	$\boxed{2}$	∞
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	$\boxed{3}$	$\boxed{2}$	4
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	$\boxed{3}$	$\boxed{2}$	4
$\boxed{0}$	$\boxed{1}$	$\boxed{3}$	$\boxed{3}$	$\boxed{2}$	$\boxed{4}$

Shortest path -
$s \to a \to c \to e$
or
$s \to a \to d \to e$

Shortest distance = 4 units

Representation of a Graph -

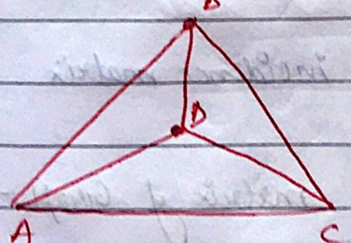
If a graph $G(V, E)$ consists of n-vertices then the adjacency matrix of a graph is an $n \times n$ matrix $A = [a_{ij}]$ and defined by -

Adjacency Matrix

$$a_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_j\} \text{ has an edge, i.e., } v_i \text{ is adjacent to } v_j \\ 0, & \text{if there is no edge b/w } v_i \text{ and } v_j. \end{cases}$$

If there exists an edge b/w vertex v_i & v_j, where i is a row and j is a column then the value of $a_{ij} = 1$. If no edge, value of $a_{ij} = 0$.

In this way we find adjacency matrix.



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}_{4\times 4} = M_A$$

This is one way of representing a graph, which contain only vertices.

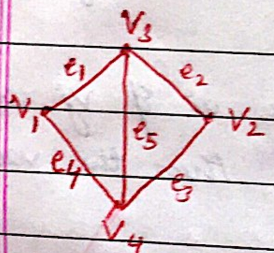
② Incidence Matrix Representation

If an undirected / directed graph $G(V, E)$ consists of 'n' vertices and 'm' edges, then incidence matrix $n \times m$ matrix $C = [c_{ij}]$ and defined by

$$c_{ij} = \begin{bmatrix} 1, \text{ if the vertex } v_i \text{ is incident by edge } e_j \\ 0, \text{ otherwise} \end{bmatrix}$$

Row for Vertex and column for edge
The no. of ones (1's) in an incidence matrix of a graph (without loops) is equal to the sum of degrees of all vertices in a graph.

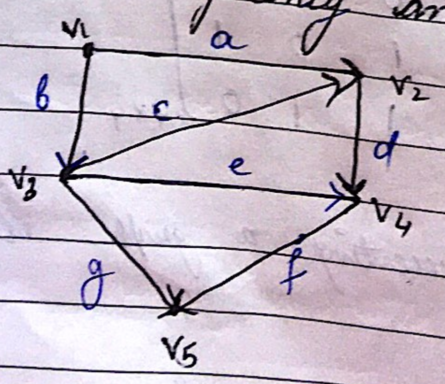
// for undirected graph



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}_{4\times5} \end{array}$$

$deg(v_1) = 2$ $deg(v_3) = 3$
$deg(v_2) = 2$ $deg(v_4) = 3$
$\Sigma deg(v) = 10$ = no. of 1's in incidence matrix.

Q Find the adjacency and incidence matrix of Graph-



adjacency // matrix

$$a_{ij} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_2 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5 \times 5} \end{array}$$

incidence matrix //

$$x_{ij} \quad \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}_{5 \times 7} \end{array}$$

Note: The value in incidence matrix of a directed graph is-

1 → for outgoing edge

0 → for non-incident ~~egdg~~ edge

-1 → for incoming edge.

PAGE NO.:
DATE: / /

Tree

Tree – (Connected graph without loops & parallel edges)
A tree is an acyclic graph having no cycles.

undirected || A tree or general tree is defined as a non-empty finite set of elements called as vertices or nodes having the property that each node can have minimum degree 1 and maximum degree 'n'.

- It represents relationship b/w Individual element or nodes.
- There is a unique path b/w every pair of vertex.
- It is a spanning graph.

Root –
The vertex with degree '0' or no path is connected to it is called root node.

leaf node –
The vertex having degree '1' is leaf node.

A tree with 'n' vertices ~~is~~ contains (n-1) edges

Parent node –
Node having a child node.

★ <u>Child node</u>

Node having a parent node

★ <u>Binary tree</u>

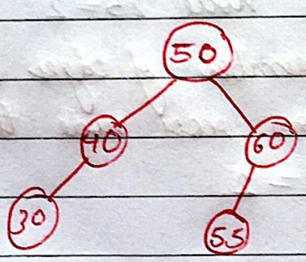
A tree in which a parent has not more than 2 children, is called a binary tree.

★ <u>Binary Search tree</u>

In BST —

X, in left subtree of vertex V, value (X) ≤ value (V)

Y, in right subtree of vertex V, value (~~Z~~ Y) ≥ value(V)



// Rooted tree + BST

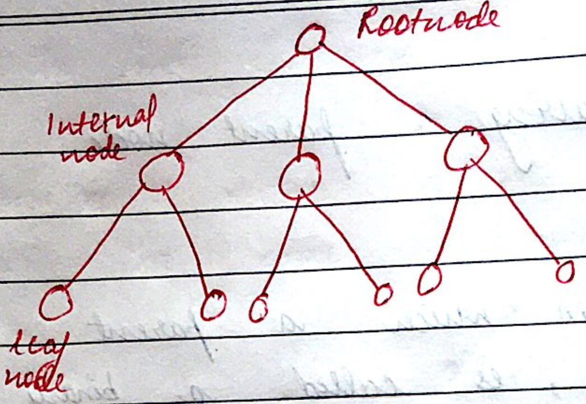
★ <u>Rooted Tree</u>

A rooted tree G is a connected acyclic graph with a special node that is called root of the tree and every edge directly or indirectly originates from the root.

★ <u>m-ary tree</u> -

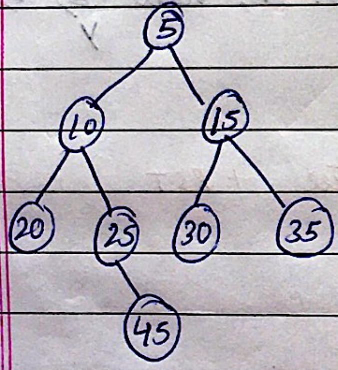
If every internal vertex of a rooted tree has not more than 'm' children, it is called m-ary tree.

If, every internal vertex of a rooted tree has exactly 'm' children, then it is called a full m-ary tree.



☆ Depth of the tree -

No. of edges from root node to the leaf node is called depth of the tree.



Depth of node 25 = 2 as, the number of edges in the path from root node to the node 25 = 2.

☆ Height of the tree :-

The no. of edges in the longest path connecting the node ~~25~~ to any leaf node.

In the above graph.

$\boxed{\text{Height of node } 25 = 1}$ as only 1 edge is connecting vertex 25 to leaf node (45)

Height of 5 = 3

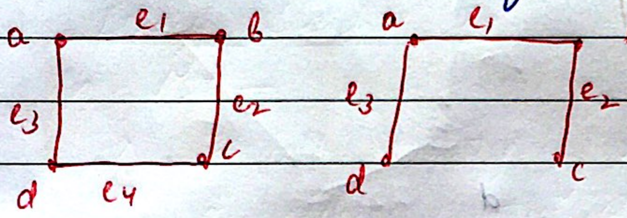
☆ Trivel and Non-Trivel

Trivel where only a single vertex exists

Non-Trivel - no. of vertices are more than one.

Spanning Tree

Let G be a graph with n-vertices and e-edges then spanning tree T of G is a subgraph of G having
a) All the vertices of G
b) It is connected subgraph of G
c) It does not have any loop or circuit



// spanning Tree {without any circuit and there exist a path b/w every pair of vertex}

Chord ~~Branch~~ of tree -

The edges of G that are dropped in spanning tree is called chord of tree (e_4)

Branch of tree

The edges of G (V, E) that ~~are~~ is included in spanning tree is called Branch of spanning tree

Rank of tree -

The no. of edges of G which are included in spanning tree ~~are~~ is the rank of the tree.
In above graph $\{e_1, e_3, e_2\}$ are branch of the tree and rank is $R(G) = 3$ // there are 3 branches

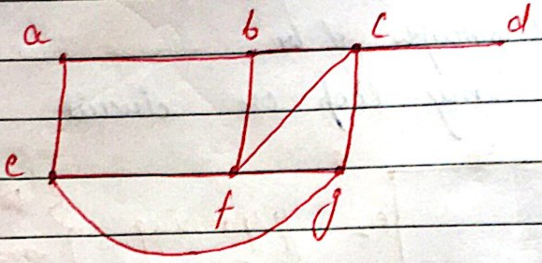
Nullity -

The no. of edges of G(V, E) that are not included in spanning tree denotes the nullity of that tree
Here, $\{e_4$ is the chord of the tree ; and nullity$(G) = 1\}$
// for 1 edge

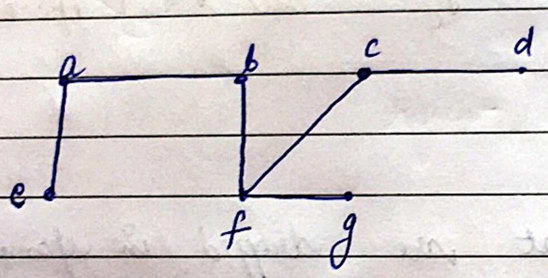
✩ Ways to find spanning tree. –

if n = no. of vertices
$(n-1)$ = no. of edges // without loops and ∥ edges

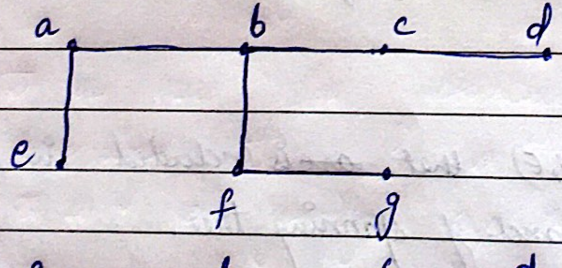
Q



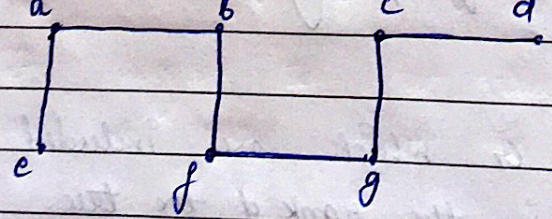
a)



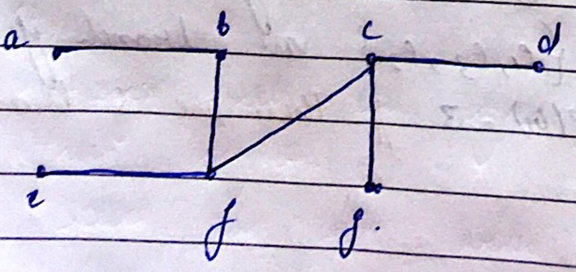
b)



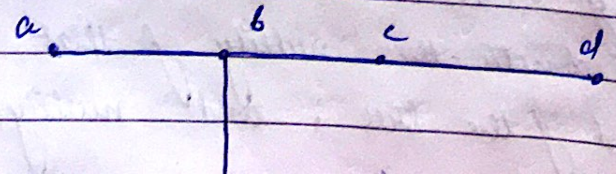
c)



d)



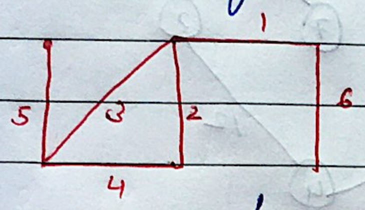
e)



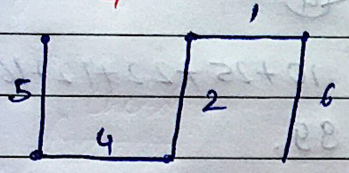
PAGE NO. :
DATE : / /

Minimal Spanning Tree

Let G be a graph in which each edge ~~are~~ is assigned a positive real number then the minimal spanning tree T of G is that spanning tree for which G have minimum length sum
(sum of the lengths of all edges is minimum)

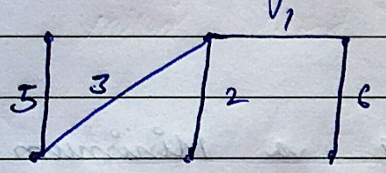
Q 

Find minimal spanning tree

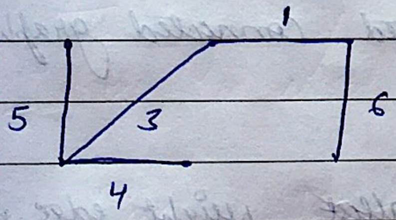


$= 5+4+2+1+6$
$= 18$

spanning tree



$= 5+3+2+1+6$
$= 17$ // minimal spanning tree



$= 5+3+4+1+6$
$= 19$

Prims Algorithm

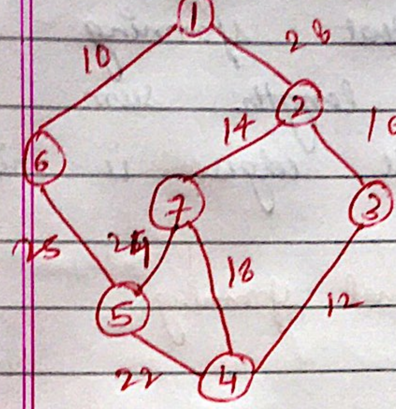
It is used to find the subset of edges that includes every vertex of the graph such that the sum of weights of edges can be minimized

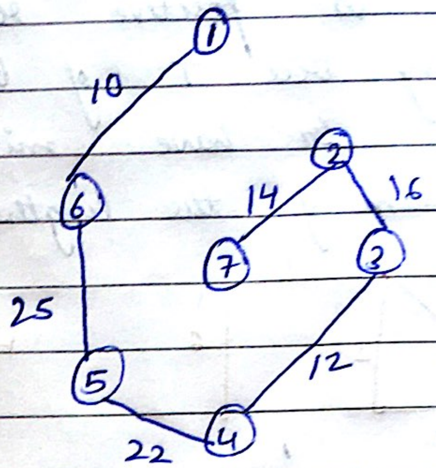
Steps to find MST using Prims algo —

i) Initialize the MST, by using a randomly choosen vertex
ii) Selected the minimum connected edge / edge with

minimum weight) to the choosen vertex.

iii) Repeat the steps until all ~~edg~~ vertices are connected.





Weight / cost = 10+25+22+12+16+14
= 99.

Krushkal Algorithm-

It is used for constructing a Minimum Spanning tree for a weighted connected graph. It is a greedy algorithm.

It suggest to put the smallest weight edge in such a way, that it does'nt forms a cycle.

Steps

i) Arrange all the edges in increasing order of their weight.

ii) Select the edge with minimum weight and plot it in the graph,

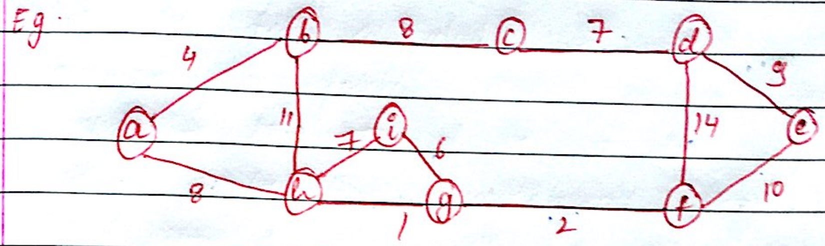
iii) Plot the edge in such a way that it should'nt form a cycle.

iv) Repeat until (n-1) edges are used where

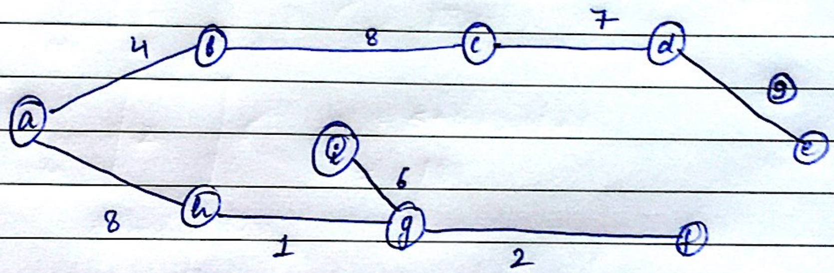
These algos don't work for disconnected graphs.

n = no. of vertices.

Eg.



$= E = \{1, 2, 4, 6, 7, 7, 8, 8, 9, 10, 11, 14\}$



$\text{cost} = 4+8+1+6+8+2+7+9$

$= 45$