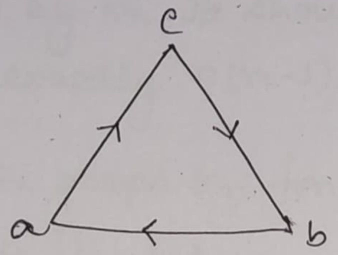
Graph Theory

Definitions

1) Simple graph :→

A graph which has neither loops nor multiple edges i.e., where each edge connects two distinct vertices and no two edges connect the same pair of vertices is called a simple graph.

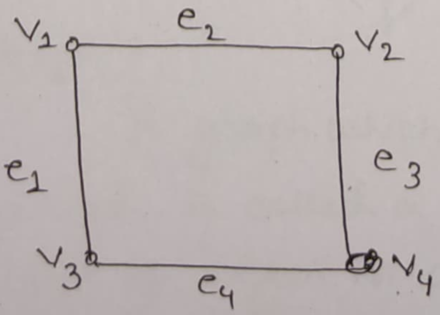
Fig - below represents simple graph.



2) Regular graph :→

A graph in which all vertices are of equal degree, is called a regular graph.

If the degree of each vertex is r. then the graph is called a regular graph of degree r.



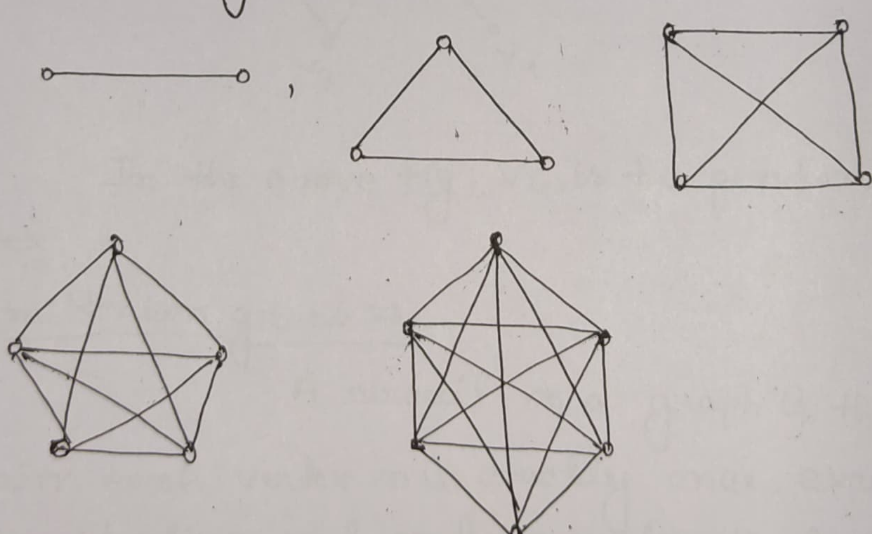
Regular graph

3) Complete graph :-

A simple graph G is said to be complete if every vertex in G is connected with every other vertex, ie., if G contains exactly one edge between each pair of distinct vertices.

A complete graph is usually denoted by K_n. It should be noted that K_n has exactly $n(n-1)/2$ edge.

The graps K_n for $n = 1, 2, 3, 4, 5, 6$ shown in fig below.

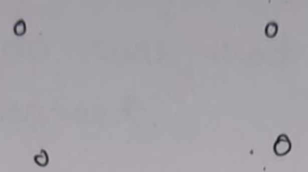


4) Null graph :-

A graph which contains only isolated node, is called a null graph. i.e., the set of edges in a null graph is empty.

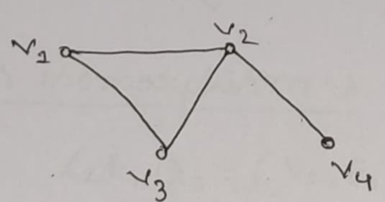
Null graph is denoted on n vertices by N_n

N_4 is shown in fig below. note that each vertex of a null graph is isolated.



5) <u>Pendent or end vertex :-</u>

A vertex of degree one, is called a pendent vertex or an end vertex.



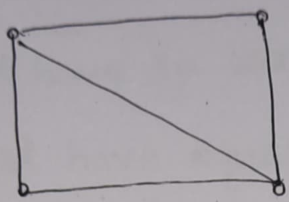
In the above fig. V_4 is the pendent vertex.

6) <u>Hamiltonian graps :-</u>

A circuit in a graph G that contains each vertex in G exactly once, except for the starting and ending vertex that appears twice is known as Hamiltonian circuit.

A grap G is called a Hamiltonian graph if it contains a Hamiltonian circuit.

6) **planner graph** A graph G is said to be planner if there exists some geometric representation of G which can be drawn on a plane such that no two of its edges ~~s~~ are intersect.



The above fig. is a planner graph.

7) **Graphs isomorphism :→**

Let, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. A function $f : V_1 \longrightarrow V_2$ is called a graphs isomorphism if

i) f is one-to-one and onto.

ii) for all $a, b \in V_1$, $\{a, b\} \in E_1$ if and only if $\{f(a); f(b)\} \in E_2$. when such a function exists G_1 and G_2 are called isomorphic graps and is written as $G_1 \cong G_2$

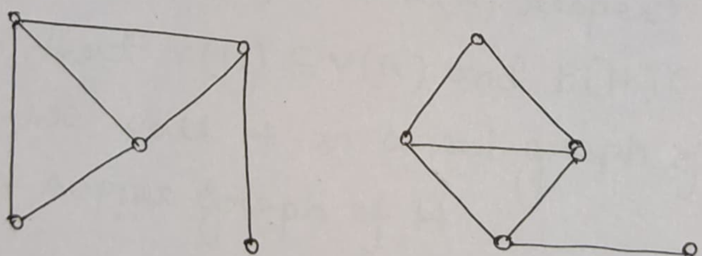
In other words. ~~#~~ two graphs G_1 and G_2 ared said to be isomorphic to each other if there is a one-to-one correspondence between their vertices and between edges such that ~~incident~~

incident relationship. ϕ is preserve
written as $G_1 \cong G_2$ or $G_1 = G_2$

The necessary conditions for two graphs to be isomorphic are:-

1. Both must have the same number of vertices.
2. Both must have the same number of edges.
3. Both must have equal number of vertices with the same degree.
4. They must have the same degree sequence and same cycle vector $(c_1, \cdots c_n)$ where c_i is the number of cycles of length i,

Eg.

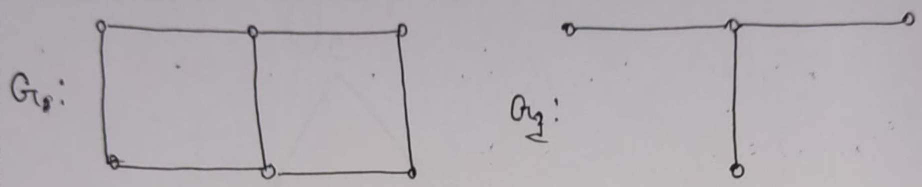


The above two fig(s) are isomorphic to each other. because.

i) They both have 5 vertices.
ii) They both have 6 edges.
iii) equal number of vertices with same degree
iv) They both have same degree sequence.

8) **SUB graph :→**

A subgraph of G is a graph having all of its vertices and edges in G. if G_1 is a sub graph of G_2, then G is a supper graph of G_1.



G_1 is a sub graph of G.

In other words. If G and H are two graphs with vertex sets V(H), V(G) and edge sets E(H) and E(G) respectively such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ then we call H as a sub graph of G or G as a supper graph of H.

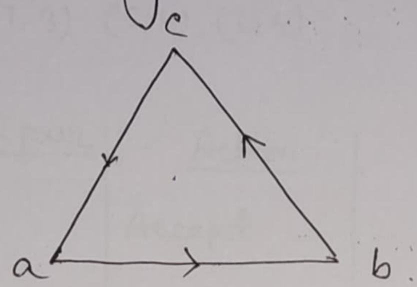
9) **Digraph :→**

A directed graph or digraph G consists of a set ~~of~~ V of vertices and a set E of edges such that $e \in E$ is associated with an ordered pair of vertices.

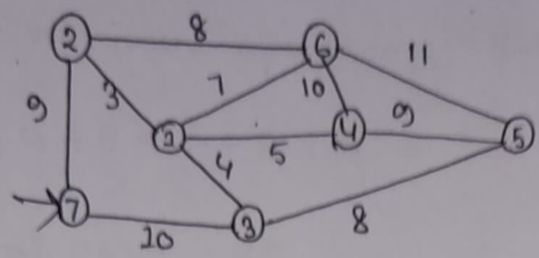
In other words, if each edge of the graph G has a ~~directed~~ direction then the graph is called directed graph

In the diagram of directed graph each edge $e = (u, v)$ is represented by an arrow or directed curve from initial point u of e to the terminal point v.

The following fig. is an example of a directed graph.



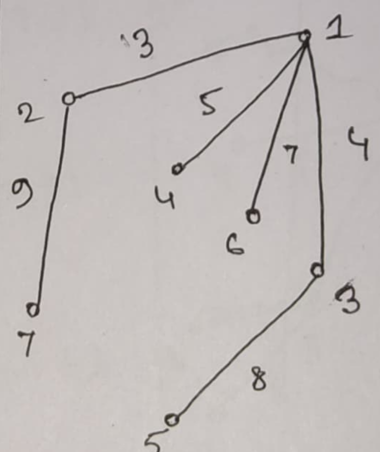
Year-2009



Krushkal

$\overset{3}{(2,1)}$ $\overset{4}{(1,3)}$ $\overset{5}{(1,4)}$ $\overset{7}{(1,6)}$ $\overset{8}{(3,5)}$ $\overset{9}{(2,7)}$ $\overset{9}{(4,5)}$
$\overset{8}{(2,6)}$ $\overset{10}{(7,3)}$ $\overset{10}{(6,4)}$ $\overset{11}{(6,5)}$

Selected pair	Action	Spanning tree
(2, 1)	Accept	
(1, 3)	Accept	
(1, 4)	Accept	
(1, 6)	〃	
(3, 5)	〃	
(2, 7)	〃	
(4, 5)	Reject	
(2, 6)	〃	
(7, 3)	〃	
(6, 4)	〃	
(6, 5)	〃	



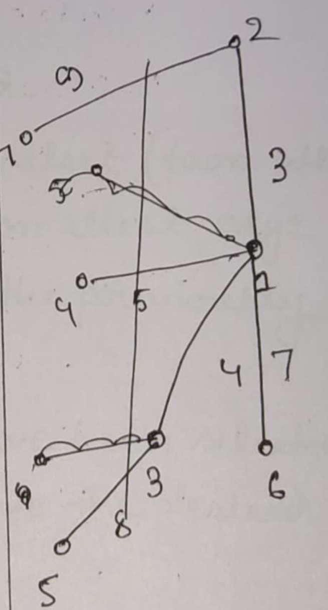
Cost of minimul spanning tree

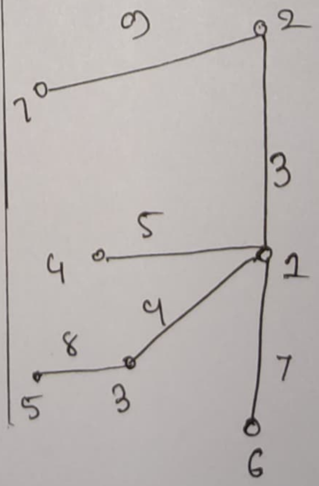
$= 3 + 4 + 5 + 7 + 8 + 9 = 36$

$\therefore$ Total cost $= 36$.

<u>prim's algorithm</u>

Starting/ Choosed vertex	vertex sets	selected set	spanning tree
7	(2,7) (7,3)	(2,7)	
2, 7	(7,3) (2,6) (2,1)	(2,1)	
2, 7, 1	(7,3) (2,6) (1,3) (1,6) (1,4)	(1,3)	
2, 7, 1, 3	~~(7,3)~~ (2,6) (1,6) (1,4) (3,5)	(1,4)	
2, 7, 1, 3, 4	~~(1,3)~~ (2,6) (1,6) (3,5) ~~(6,4)~~ (4,5)	(1,6)	
2, 7, 1, 3, 4). 6	~~(2,6)~~ (3,5) (4,5) (6,5)	(3,5)	
2, 7, 1, 3, 4, 6, 5			





Total cost $= 3 + 4 + 5 + 7 + 9 + 8 = 36$.

Year → 2022

B) b) Describe the kruskal's algorithm for finding the minimum spanning tree for a graph G. Comment about the complexity of the algorithm.

Ans There are several method available for actually finding a shortest spanning tree in a given graph. One algorithm is kruskal is as fallows —

i) List all edges of the graph G in order of nondecreasing weight.

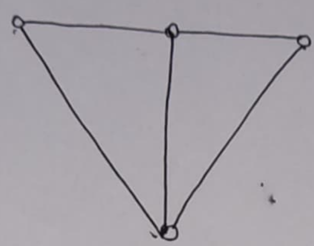
ii) Next, select a smallest edge of G.

iii) Then for each successive step select (from all remaining edges of G) another smallest edge that makes no circuit with the previously Selected edges.

iv) Continue until (n - 1) edges have been selected and these edges will constitute the desired shortest spanning tree.

Year - 2012

o) Find all the spanning trees of the Graph shown below.



How many spanning tree can be formed? Why?

n^{n-2} $\quad 4^{4-2}$

$4^2 \quad \underline{16}$

year - 2013

2) b) What is Minimal Spanning Tree? Explain Prim's algorithm with non-trivial illustration.

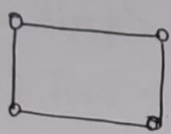
6) What is the purpose of Floyd's algorithm? Explain its working principle. Derive the worst case time complexity of this algorithm.

Path graphs and cycle graphs :→

A connected graph that is two regular is called a cycle graph.

[two regular means each vertex has degree 2]

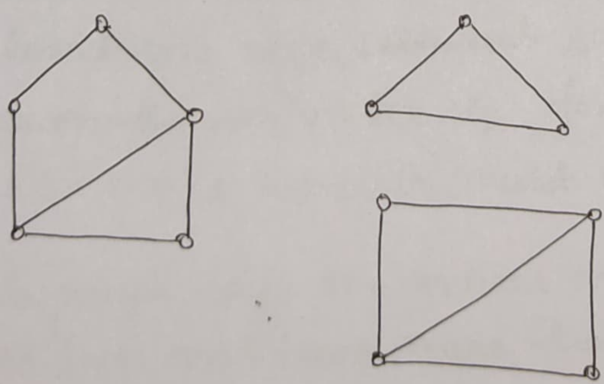
Eg:



In the cycle graph G, by removing an cycle is called path graph.

** What is rank and what is Nullity :→

For a graph G with n vertices, m edges and k components.



we define the rank of G.

$$\boxed{\text{rank} = \rho(G) = n - k}$$

and nullity defines that,

$$\mu(G) = m - \rho(G)$$

$$\boxed{\mu(G) = m - n + k}$$

If G is connected then $K = 1$

$\boxed{\text{Then, rank} = n-1}$

$\boxed{\text{Nullity} = m-n+1}$

*

1) Prove that, a simple graph with n-vertices must be connected if it has more than $(n-1)(n-2)/2$ edges.

Ans Consider, a simple graph, with n vertices. Choose $(n-1)$ vertices i.e., $v_1, v_2, \dots v_n$ of G, we have maximum edges with $(n-1)$ vertices will be -

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

Thus, we have, more than $(n-1)(n-2)/2$ edges. at least one edge ~~without~~ should be drawn between the nth vertex V_n for two some V_i. where $1 \leq i \leq n-1$ hence G must be connected.

**

2) In a simple graph with n-vertices and K components can not have more than, $(n-K)(n-K+1)/2$ edges.

Ans Let, n_i be the no. of vertices in ith component so therefore,

$$\sum_{i=1}^{K} n_i = n.$$

i.e, $n_1 + n_2 + n_3 + \dots + n_K = n$

A component with n_i vertices have maximum possible no. of edges when it is complete. i.e, It will contain $\frac{n_i(n_i-1)}{2}$ edges.

Hence, maximum no. of edges is

$$\frac{1}{2}\sum_{i=1}^{k} n_i(n_i-1)$$

$$= \frac{1}{2}\sum_{i=1}^{k} n_i^2 - \frac{1}{2}\sum_{i=1}^{k} n_i$$

$$\leq \frac{1}{2}[n^2-(k-1)(2n-k)] - \frac{1}{2}n$$

[from the following algebric inequality]

$$\leq \frac{1}{2}[n^2-2nk+k^2+n-k]$$

$$\leq \frac{1}{2}(n-k)(n-k+1)$$

The following inequality can be solved by -

$$\sum_{i=1}^{k}(n_i-1) = n-k$$

and squaring both side we get -

$$\sum_{i=1}^{k}(n_i-1)^2 = (n-k)^2$$

$$\Rightarrow \sum_{i=1}^{k} n_i^2-2n_i+1+$$

incomplete

* Let, G be a disconnected graph with n vertices where n is even. if G has two components each of which is complete then G has minimum

$$\frac{n(n-2)}{4} \text{ edges.}$$

$\underline{m}$ Let, x be the no. of vertices in one of the component. then, the other component has $(n-x)$ no. of vertices. Since, both components are complete then, the no. of edges they have $\frac{x(x-1)}{2}$ and $\frac{(n-x)(n-x-1)}{2}$ respectively.

if m be the total no. of edges then,

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

$$= \frac{x^2 - \cancel{x} + n^2 - nx - n - nx + x^2 + \cancel{x}}{2}$$

$$= \frac{2x^2 - 2nx + n^2 - n}{2}$$

$$= x^2 - nx + \frac{n^2}{2} - \frac{n}{2}$$

$$m = x^2 - nx + \frac{n}{2}(n-1)$$

$$\frac{dm}{dx} = 2x - n$$

$$\frac{d^2m}{dx^2} = 2 > 0$$

therefore, m is minimum.

$$2x - n = 0$$

$$\Rightarrow x = n/2$$

$$m = x^2 - nx + \frac{n}{2}(n-1)$$

$$= \left(\frac{n}{2}\right)^2 - \frac{n}{2} \times n + \frac{n}{2}(n-1)$$

$$= \frac{n^2}{4} - \cancel{\frac{n^2}{2}} + \cancel{\frac{n^2}{2}} - \frac{n}{2}$$

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

$$= \frac{n}{4}(n-2)$$

$$= \frac{n(n-2)}{4} \quad \text{[Proved]}$$

Q) Find the rank and nullity of a complete graph K_n.

Ans According to the problem,

$$\therefore K_n = 1$$

$$\text{Rank} = n - 1$$

$$\text{Nullity} = m - n + 1$$

for complete graph $m = \frac{n(n-1)}{2}$

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

$$= \frac{n^2 - n - 2n + 2}{2}$$

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

$$\boxed{\text{Nullity} = \frac{(n-1)(n+2)}{2}}$$

Walk:→ A walk is define as a finite alternative sequence of vertices and edges of the form

$(V_i, E_j)\ (V_{i+1}, E_{j+1})\ (V_{i+2}, E_{j+2}) \cdot \cdots (E_K, V_m)$

which begings and ends with vertices.

such that,

i) each edge in the sequence is incident on the vertices preceding and following it in the sequence.

ii) No edge appears more than 1 in the sequence is called walk on trial in G.

The vertex with which a walk begins is called initial vertex and the vertex with which the walk ends is called final vertex.

⇒ a walk that begings and ends with the same vertex is called closed walk.

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

⇒ an open walk in which no vertex appear more than one is called path.

⇒ A closed walk with atleast one edge in which no-vertex except the treminal vertex appear more than one is called cycle or circuit

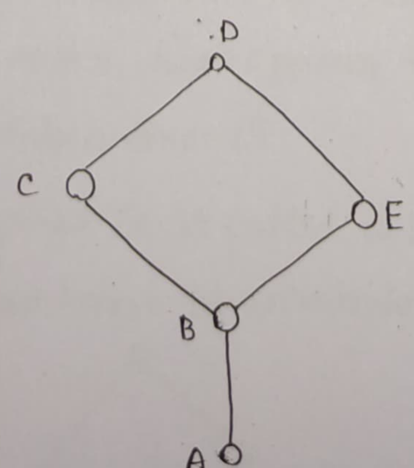
⇒ The no. of edges in a walk is called its length

Euler graph :→

A path in graph G is called Euler path. if it includes every edge exactly once. Since, the path is called Euler trail.

A Eular path that is a circuit is called Eulerian circuit.

The graph which has an Eulerian circuit is called Eulerian circuit graph.



Thearem

A connected graph G has an Eular trail if an only if it has almost two vertices. since, it has either no vertices of odd degree. or exactly two vertices of odd degree.

proof:

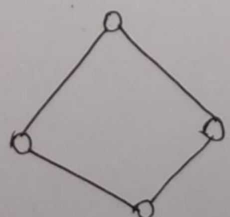
Suppose, G has an Eular trail. which is not closed. Since, each vertex in the middle of the trail is associated with two edges. Since, there is only one edge associated with end-vertex of the trail. These end vertices must be odd and other vertices must be even.

Conversly. suppose that G is connected with atmost two vertices. If G has no odd Vertices then G is Eular and so, has Euler traid.

Hammintoniah graph:-) **

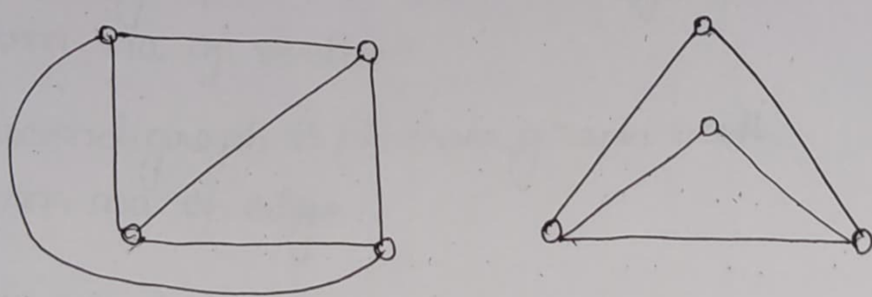
A circuit in a graph G contains each vertex in graph exactly once. except the starting and ending vertex that appear twice is known as Hamminrtonian circuit

A graph G is called a Hemmintonial graph If it contains Hammintonian circuit.



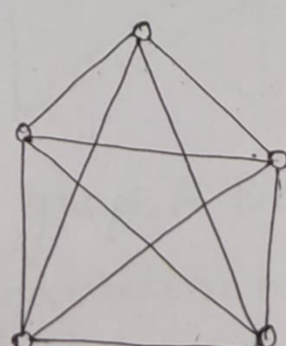
Planner graph: → A graph G said to be planner if there exists some geometric representation of G which can be drawn in a plane. i.e., no two of its edge are intersect.

The point of intersection is called cross-over.

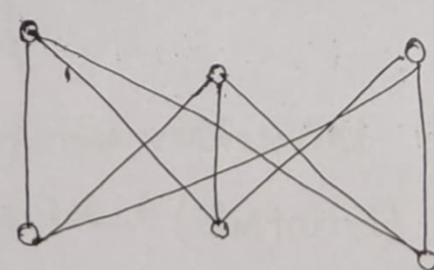


planner graph

Koratowski's graph: →



K_5



$K_{3,3}$

The complete graph with 5-vertices is the first of two graphs and they connected with 6-vertices and 9 edges are non-planner graph.

Observation

i) Both are 3-regular graph.

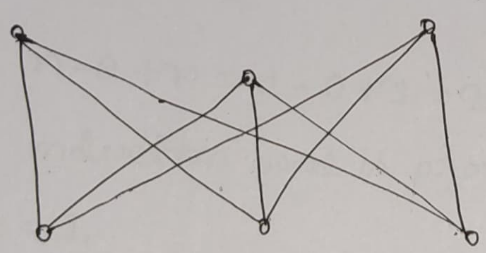
ii) Both are non-planner.

iii) Removal of one edge on one vertex makes them planner.

iv) The first graph is non-planner graph with minimum no. of vertices.

v) The second graph is planner graph with minimum no. of edge.

Bipartite graph:-) **



A graph is Bipartite if the vertex set V can be partition into two subset (disjoint) V_1 and V_2 such that, every edge e connects a vertex V_1 ~~and~~ with a vertex in V_2 (so that no edge in G connects either two vertices of V_1 ~~and~~ with two vertices of V_2.) V_1 and V_2 are called bipartition of G and the graph is called bipartite graph.

** Eular's formula

If a connected planner graph G has n-vertices e edges and r regions then.

$$\boxed{n-e+r=2}$$

This formula can be proved by Induction.

Induction base:-

If $e=0$ then. G must have one vertex so, $n=1$.

and one infinite region

i.e., $r=1$

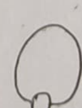
$\therefore\ n-e+r = 1-0+1 = 2$

So, Induction base is proved.

If $e=1$,

then, no. of vertices in G is either one or two.

for $n=1$, $r=2$



then, $1-1+2=2$

for, $n=2$, $r=1$

then, $2-1+1=2$

Induction hypothesis:-

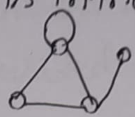
Suppose the result is true for any connected graph G with n-vertices e edges r regions. where

$$n-e+r=2$$

Induction step:-

We add one new edge to G to form a connected graph G there are the following possibilities.

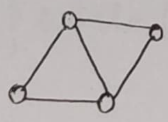
Let, k be the edge which is introduced in a graph G.



i) k is a loop, in which a new region bounded by the loop is created but the no. of vertices remains unchanged.

then, $n-(e+1)+(r+1) = n-e+r$

ii)

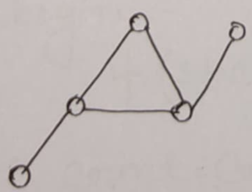


k joints two distinct vertices in which case one of the region split into two.

the no. of vertices remains unchanged.

then. $n-(e+1)+(r+1) = n-e+r$

iii)



k is incident with only one vertex of G on which case another vertex must be added. no. of region remain unchanged. pendent and increase of vertex

$=(n+1)-(e+1)+r = n-e+r$

To count the number of component of a graph using DFS.

The depth first search mechanism use a stack for graph traversal if maintain a boolean vector of size n. if there are n-vertices available each of the vertex with in the vector will be identified by 1. if it is visited or 0 otherwise.

```
do
{
  if (stack empty && not all vertex visited)
  Count = Count + 1 / Increment component.
endif
} while (stack ! Empty OR ! all vertex visited);
```

performing DFS traversal starting from the arbitary vertex v.

2. Repeat while (the stack is not empty ~~&&~~ ||
 the vector is not marked by 1)
 (not all vertex visited)

begin
 if (stack is empty && ! all vertex visited)

3. Set, count = Count + 1; // to count the no. of component.

endif

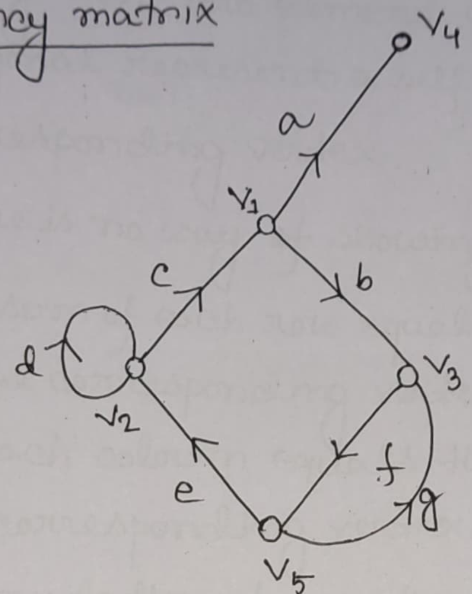
4. Choose any arbitrary vertex from the remaining vertex.

5. Repeat the above step.

6. end while

7. end procedure.

Adjacency matrix



	V_1	V_2	V_3	V_4	V_5
V_1	0	0	1	1	0
V_2	1	1	0	0	0
V_3	0	0	0	0	1
V_4	0	0	0	0	0
V_5	0	1	1	0	0

The adjacency matrix is used in representing the di-graph $x_{ij} = 1$. if there is a edge directed from ith vertex to jth vertex. $x_{ij} = 0$ otherwise.

Observation

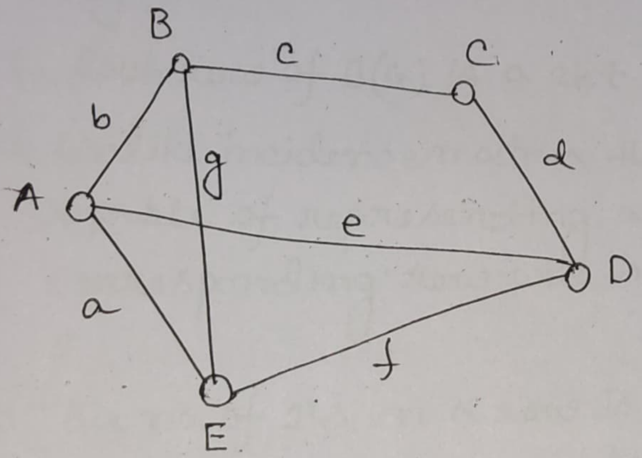
1. x is symmetric matrix if and only if the graph is a symmetric di-graph [The matrix will be symmetric for an un-directed graph].
2. Every non-zero element on the main diagonal represent a self-loop at the corresponding vertex.
3. There is no way of showing parallel edge.
4. The sum of each row equals the out degree of the corresponding vertex and the sum of each column equals the in-degree of the corresponding vertex.
5. permudation of any two rows accompained by a permudation of corresponding colums does not alter ~~after~~ the graph.
6. Permudation ~~m~~ nearly corresponds to a re-ordering of verticas. Thus two di-graph is ~~m~~ isomorphic if and only if their ~~aje~~-adjecency matrices differ only by such permudation.
7. Unlike incidence matrix the ckt matrix is capable of representing

if x is a adjecency then the tranpose matrix x^T is the adjecency matrix for a di-graph obtained by reversing the direction of every edge in G.

8. For any square matrix of order n there exist a unique di-graph G of n-vertices such that, the adjacency matrix of G.

$$X = \left[\begin{array}{c|c} X_1 & 0 \\ \hline 0 & X_2 \end{array}\right]$$

<u>Circuit matrix:→ **</u>



1. ABE, 2. ABCD 3. BCED 4. ABCDE 5. ADE

	a	b	c	d	e	f	g
1	1	1	0	0	0	0	1
2	0	1	1	1	1	0	0
3	0	0	1	1	0	1	1
4	1	1	1	1	0	1	0
5	1	0	0	0	1	1	0

Let, no different ckt in a graph G be q and no of edges in G be e then the ckt matrix b_{ij} is $q \times e$ $(0, 1)$ matrix.

$$B_{ij} = 1. \text{ if } i\text{th ckt includes } j\text{th edge}$$
$$= 0 \text{ otherwise.}$$

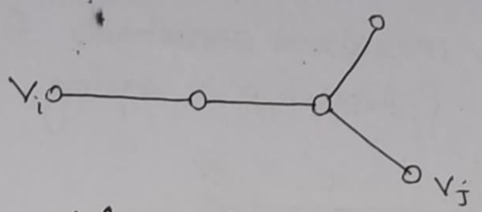
<u>The following observation can be made</u>

1. The coloum of all 0's correspond to a nonckt edge.
2. Each row of $B(G)$ is a ckt vector.
3. Unlike incidence matrix the ckt matrix is capable of representing a self loop. the corresponding row will have corresponding 1.
4. The no. of 1's in a row is equal to the no. of edge in the corresponding ckt.
5. If graph G is seperable or disconnected and consist of two blocks (or components) G_1 and G_2 then the ckt matrix $B(G)$ can be written in a block diagonal form.

$$B_{ij} = \left[\begin{array}{c|c} B(G_1) & 0 \\ \hline 0 & B(G_2) \end{array}\right]$$

Distance of a graph :→ **

In a connected graph G the distance $d(V_i, V_j)$ between two of its vertices V_i and V_j is the length of the sortest path. that is, the no. of edges.



$d(V_i, V_j) = 3.$

What is metric :→ **

A function $f(x,y)$ of two variables (a distance between them) this must satisfy the following criteria to become a metric.

1) Non - negativity

$f(x,y) \geq 0$ & $f(x,y) = 0$ if and only if $x = y$.

2) Symmetry

$f(x,y) = f(y,x)$

3) Triangle in-equality.

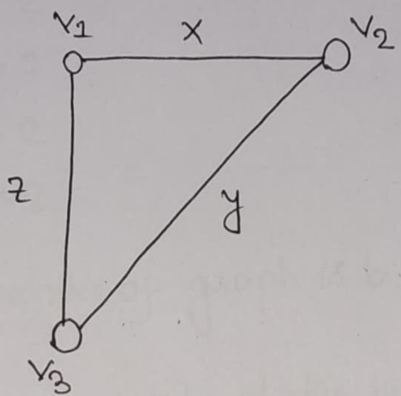
$f(x,y) \leq f(x,z) + f(z,y)$ for any z.

A function that satisfies this condition is called metric. that the distance $d(V_i, V_j)$ between two vertices of a connected graph satisfies condition 1 and 2 is immediately evident. Since $d(v_i, v_j)$ is the length of sortest path between v_i and v_j. this path cannot longer than path between v_i and v_j which goes through a specified vertex v_K.

$$d(v_i, v_j) \leq d(v_i, v_K) + d(v_K, v_j)$$

[N.B distance between vertices of a connected graph is a metric]

Eg.



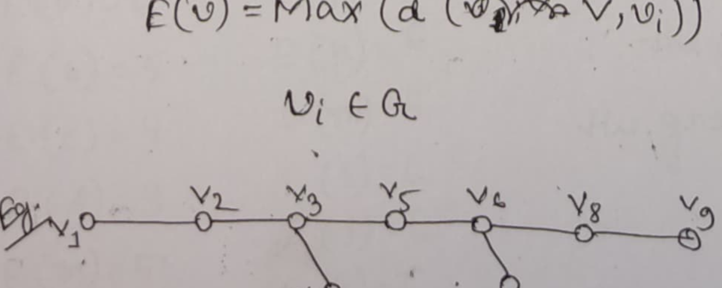
$$d(v_1, v_2) + d(v_2, v_3) \geqslant d(v_1, v_3)$$

*# Essentricity :-

It also refered to as associated number or seperation of a vertex in a graph. The essentricity $E(v)$ of a vertex in a graph G is the distance from to the vertex furthest from v in G. i.e.,

$$E(v) = \mathrm{Max}\,(d(v, v_i)) \quad v_i \in G$$

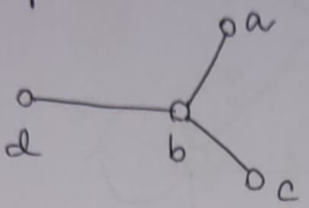
Eg.



$d(v_1, v_2) = 1, \quad E(v_1) = v_9 = 6$

** **center :→** A vertex with minimum essentricity in a graph G is called center of G.

Eg.



for the above graph the essentricity of 4 vertices are.

$E(a) = 2$

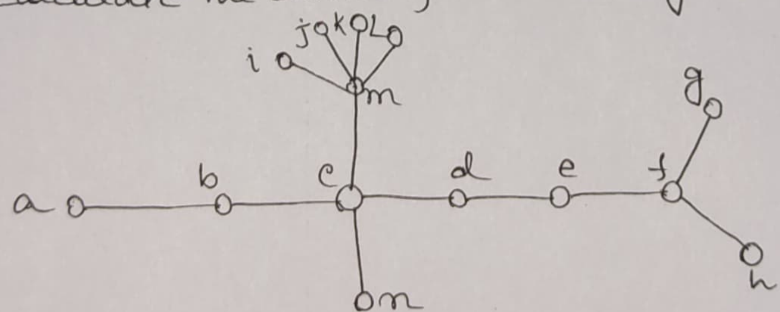
$E(b) = 1$

$E(c) = 2$

$E(d) = 2$

So, the center of graph is b.

**
Calculate the center of the below graph.

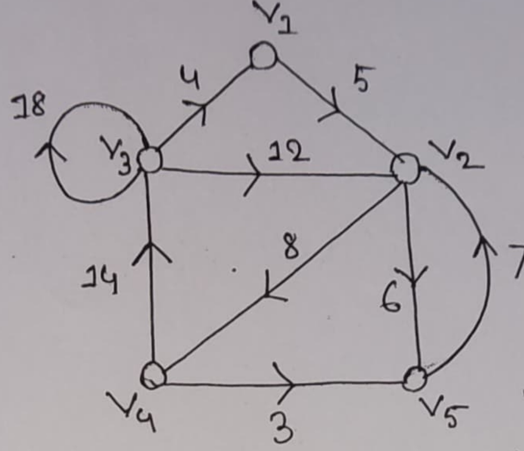


Ans for the above graph the essentricity of vertices are –

$E(a) = 6$	$E(g) = 6$
$E(b) = 5$	$E(h) = 6$
$E(c) = 4$	$E(m) = 5$
$E(d) = 3$	$E(i) = 6$
$E(n) = 5$	$E(j) = 6$
$E(e) = 4$	$E(k) = 6$
$E(f) = 5$	$E(L) = 6$

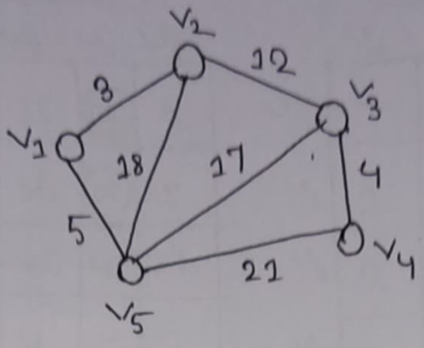
∴ the center of the graph is d

Dijikstra algorithm



{v.

Floyed alogorithm



Iteration-0

Source

	v1	v2	v3	v4	v5
v1	-	3	∝	∝	5
v2	3	-	12	∝	18
v3	∝	12	-	4	17
v4	∝	∝	4	-	21
v5	5	18	17	21	-

Destination

	v1	v2	v3	v4	v5
v1	-	2	3	4	5
v2	1	-	3	4	5
v3	1	2	-	4	5
v4	1	2	3	-	5
v5	1	2	3	4	-

Iteration-1

Source

	v1	v2	v3	v4	v5
v1	-	3	∝	∝	5
v2	3	-	12	∝	8
v3	∝	12	-	4	17
v4	∝	∝	4	-	21
v5	5	8	17	21	-

Destination

	v1	v2	v3	v4	v5
v1	-	2	3	4	5
v2	1	-	3	4	1
v3	1	2	-	4	5
v4	1	2	3	-	5
v5	1	1	3	4	-

Iteration - 2

Source

	v_1	v_2	v_3	v_4	v_5
v_1	–	3	15	α	5
v_2	3	–	12	α	8
v_3	15	12	–	4	17
v_4	α	α	4	–	21
v_5	5	8	17	21	–

Destination

	v_1	v_2	v_3	v_4	v_5
v_1	–	2	2	4	5
v_2	1	–	3	4	1
v_3	2	2	–	4	5
v_4	1	2	3	–	5
v_5	1	1	3	4	–

Iteration - 3

Source

	v_1	v_2	v_3	v_4	v_5
v_1	–	3	15	19	5
v_2	3	–	12	16	8
v_3	15	12	–	4	17
v_4	19	16	4	–	21
v_5	5	8	17	21	–

Destination

	v_1	v_2	v_3	v_4	v_5
v_1	–	2	2	3	5
v_2	1	–	3	3	1
v_3	2	2	–	4	5
v_4	3	3	3	–	5
v_5	1	1	3	4	–

Iteration - 4

Source

	v_1	v_2	v_3	v_4	v_5
v_1	–	3	15	19	5
v_2	3	–	12	16	8
v_3	15	12	–	4	17
v_4	19	16	4	–	21
v_5	5	8	17	21	–

Destination

	v_1	v_2	v_3	v_4	v_5
v_1	–	2	2	3	5
v_2	1	–	3	3	1
v_3	2	2	–	4	5
v_4	3	3	3	–	5
v_5	1	1	3	4	–

Iteration-5

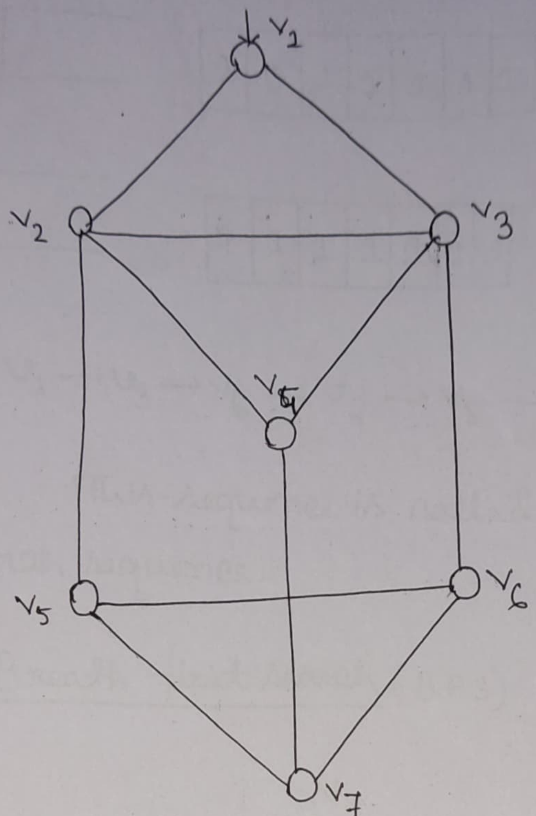
Source

	v_1	v_2	v_3	v_4	v_5
v_1	–	3	15	19	5
v_2	3	–	12	16	8
v_3	15	12	–	4	17
v_4	19	16	4	–	21
v_5	5	8	17	21	–

Destination

	v_1	v_2	v_3	v_4	v_5
v_1	–	2	2	3	5
v_2	1	–	3	3	1
v_3	2	2	–	4	5
v_4	3	3	3	–	5
v_5	1	1	3	4	–

Depth first search algorithm



DFS (stack)

Stack	Vector	Visited
V_1	0 0 0 0 0 0 0	—
	V_1 V_2 V_3 V_4 V_5 V_6 V_7	
V_2 V_3	1 0 0 0 0 0 0	V_1
V_2 V_4 V_6	1 0 1 0 0 0 0	V_3
V_2 V_4 V_5 V_7	1 0 1 0 0 1 0	V_6
V_2 V_4 V_5	1 0 1 0 0 1 1	V_7

v_2	v_4		

1	0	1	0	1	1	1

$\boxed{v_5}$

v_2		

1	0	1	1	1	1	1

$\boxed{v_4}$

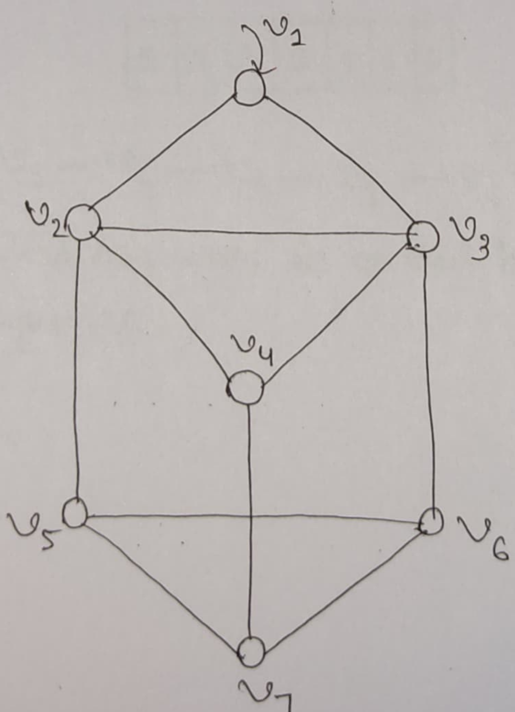
1	1	1	1	1	1	1

$\boxed{v_2}$

$$v_1 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2$$

This sequence is called depth first search sequence.

Breath first search (B.F.S)



v_1		

0	0	0	0	0	0	0

$\boxed{-}$

v_2	v_3		1	0	0	0	0	0	0	v_1
v_3	v_4	v_5	1	1	0	0	0	0	0	v_2
v_4	v_5	v_6	1	1	1	0	0	0	0	v_3
v_5	v_6	v_7	1	1	1	1	0	0	0	v_4
v_6	v_7		1	1	1	1	1	0	0	v_5
v_7			1	1	1	1	1	1	0	v_6
Empty			1	1	1	1	1	1	1	v_7

$$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7$$

This sequence is called breadth first search sequence.

INDEX

1)

Year → 2015

a) State the and prove the necessary and sufficient conditions for a graph G to be an Euler graph. (2+4)

Ans.

(2)

2) b) prove that a simple graph with n-vertices and k components can have atmost $(n-k)(n-k+1)/2$ edges. — 3 **

Ans Let the number of vertices in each of the k-components of a graph G be $n_1, n_2, \ldots, n_k$. Thus we have

$$n_1 + n_2 + n_3 + \cdots + n_k = n,$$

$$n_i \geq 1$$

The proof of the theorem depends on an algebric in-equality.

$$\sum_{i=1}^{k} n_i^2 \leq n^2 - (k-1)(2n-k) \quad \cdots \quad (1)$$

Now, the maximum number of edges in the ith component of G (which is a simple connected graph) is $\frac{1}{2} n_i (n_i - 1)$.

$\therefore$ The maximum no. of edges in G is

$$\frac{1}{2} \sum_{i=1}^{k} (n_i - 1) n_i = \frac{1}{2}\left(\sum_{i=1}^{k} n_i^2\right) - n/2$$

$$\leq \frac{1}{2}\left[n^2 - (k-1)(2n-k)\right] - \frac{n}{2}$$

[from 1]

$$= \frac{1}{2}(n-k)(n-k+1)$$ [Proved]

(3)

c) Write an algorithm for BFS ~~A~~ traversal of a graph. How will you find the no. of connected components of a given graph by the above algorithm. — 5+2

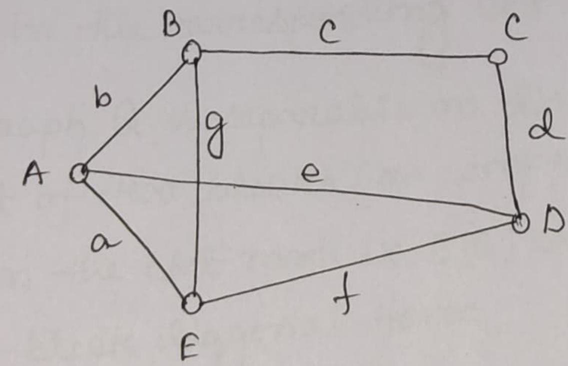
(4)

3) a) Define a circuit matrix B for G. Illustrate with a proper example. List some of vital observation that can be made about the circuit matrix B(G). — 2+4+2

Ans. **Circuit matrix:** → Let, no. of different ckt in a graph G be q and no. of edges in G be e then the ckt matrix B_{ij} is $q \times e$ (0, 1) matrix.

$$B_{ij} = 1 \text{ if } i\text{th circuit includes } j\text{th edge.}$$
$$= 0 \text{, otherwise.}$$

Eg.



Circuit

1. ABE 2. ABCD 3. BCED 4. ABCDE 5. ADE

	a	b	c	d	e	f	g
1	1	1	0	0	0	0	1
2	0	1	1	1	1	0	0
3	0	0	1	1	0	1	1
4	1	1	1	1	0	1	0
5	1	0	0	1	1	0	0

(5)

The following observation can be made

1) The coloum of all 0's correspond to a non ckt edge.

2) Each row of $B(G)$ is a ckt vector.

3) Unlike incidence matrix the ckt matrix is capable of representing a self-loop. The corresponding row will have corresponding 1.

4) The no. of 1's in a row is equal to the no. of edges in the corresponding ckt.

5) If graph G is separable or disconnected and consist of two blocks (or components) G_1 and G_2 then the ckt matrix $B(G)$ can be written in a block diagonal form.

$$B_{ij} = \left[\begin{array}{c|c} B(G_1) & 0 \\ \hline 0 & B(G_2) \end{array}\right]$$

5) c. Define a minimum spaning tree for a given un weighted un-directed graph G. Describe the Kruskal's algorithm. Clearly state your assumptions. → 2+6

Ans. Minimum spanning tree :→ A spanning tree with the smallest weight in a weighted graph is called a shortest spanning tree or shortest-distance spanning tree or minimum spanning tree.

Kruskal's algorithm

Input:→ A connected graph G with non-negative values assigned to each edge.

Output:→ A minimal spanning tree for G.

Let, $G = (V, E)$ be graph and $S = (V_s, E_s)$ be the spanning tree to be found from G. Let $|V| = n$. and $E = \{e_1, e_2, \ldots e_m\}$. The step-wise algorithm is given below.

Method

Step 1:→ Select any edge of minimal value that is not a loop. This is the first edge of E_s. if there is more than one edge of minimum value. arbitrarily choose one of these edges.

i.e., select an edge e_1 from E such that e_1 has least weight. Replace $E = E - \{e_1\}$. and

$E_S = \{e_1\}$

Step-2:→ Select any remaining edge of G having minimum value that does not from a circuit with the edges already included in E_s.

i.e., Select an edge e_i from E such that e_i has least weight and that does not from a cycle with members of E_s

set $E = E - \{e_i\}$ and $E_S = E_S \cup \{e_i\}$

<u>Step-3 :→</u> Continue step-2 until T_S contains $(n-1)$ edges. where n is the no. of vertices of G.
i.e., repeat step-2 untill $|E_S| = |V| - 1$.

Year → 2024

2(a) When are two graphs said to be isomorphic? – 2

Ans The two graphs are said to be isomorphic if the following four conditions are satisfied.

1) Both must have same no. of vertices.

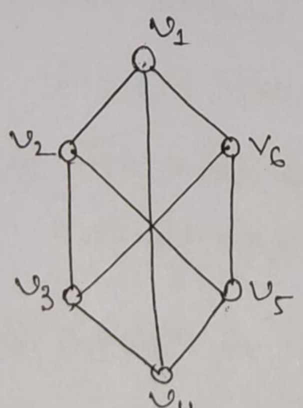
ii) Both have same no. of edges.

iii) Both must have equal no. of vertices with same degree.

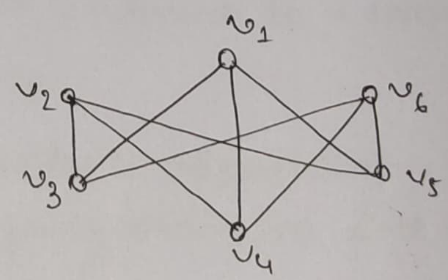
iv) They must have the same degree sequence and same cycle vector $(c_1, \cdots, c_n)$

where c_i is the no. of cycle of length i.

Eg.



G_1



G_2

In the above two graphs we have to check that the above four conditions are satisfied or not –

1) Both have 6 vertices.

ii) Both have 6 edges.

iii) Both have equal no. of vertices with same degree. i.e., degree 3.

iv) They ~~the~~ have same degree sequence and same cycle vector i.e., 5.

So, we can say that the above two graphs are isomorphic.

b) Explain in brief the Dijiktra's algorithm using pseudo-code on flow chart. — 8

To find the sortest path from a to vertex e in a waited graph carry out following procedure.

1. Assign to 'a' the label 0.

2. Until e is labeled or no furthe label can design. do the following.

   i) for each label vertex $u(x, d)$ and for each unlabel vertex v adjecent to u compute $d + w(e)$ where $e = uv$

   ii) for next label vertex u. adjecent to unlabel vertex v giving minimum $d = d + w(e)$

   assign to v the label (u, d)

   If a vertex can be label (x, d') for various vertex x. makes any choice.

3. end procedure.

(9)

c) prove that a simple graph with p vertices must be connected if it has more than $[(p-1)(p-2)]/2$ edges. — 4 **

Ans. Consider a simple graph with p - vertices.

choose $(p-1)$ vertices i.e., $v_1, v_2 \cdots v_{p-1}$ of G.

we have maximum edges, with $(p-1)$ vertices (maximum) edges will be —

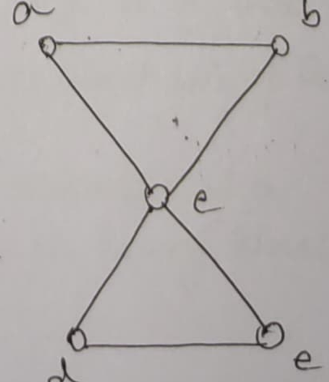
$$(p-1)(p-2)/2$$

Thus we have more than $(p-1)(p-2)/2$ edges. atleast one edge should be drawn between the pth vertex up to/into some v_i where $1 \leq i \leq p-1$.

Hence, G must be connected.

d) What is arbitrarily traceable graph? Illustrate with the help of diagram. — 2

Ans. Suppose we start from a vertex 'a'. and trace a path abc. Now at c we have a choice of going a, d on e. if we took the 1st choice we could only trace the ckt. abca which is not an eular line. Thus. starting from a we can not trace entair eular line sp simply by



5)

moving along any edge that has not already been traversed.

A vertex v ~~is~~ in an eular graph have seen that an eular line is always obtained when one follows any walk from vertex v according to the single rule. That is, whenever one arrives at a vertex one shall select any edge.

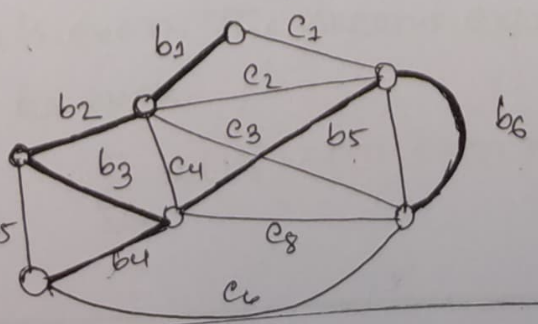
Such a graph is called an arbitarily traceable graph from v.

Here in above fig. the graph is arbitarily traceable from vertex c but not from any other vertex.

a) Define spanning tree. How many spanning ~~tree~~ ~~tree~~ trees exist in a ~~st~~ simple graph with n-vertices. — 2

→m Spanning tree :→ A tree T is said to be spanning tree of a connected graph G if T is a sub-graph of G and T contains all vertices of G.

Eg. The following graph is the example of a spanning tree. The sub-graph in heavy lines in fig is a spanning tree.



3) b) prove that the number of odd degree vertices in a simple graph is always even. — 3

Ans Let us consider a graph G. the sum of the degrees of all vertices is —

$$\sum_{i=1}^{n} d(v_i) \text{ —— (i)}$$

If we consider the vertices with odd and even degrees seperately. then quantity in the above equation can be expressed as the sum of two sums. each taken over vertices of even and odd degrees. respectively. as follows —

$$\sum_{i=1}^{n} d(v_i) = \sum_{even} d(v_j) + \sum_{odd} d(v_k) \text{ —— (ii)}$$

Since the left hand side of eq-(ii) is even and the first expression on the right-hand side is even. The second expression must also be even

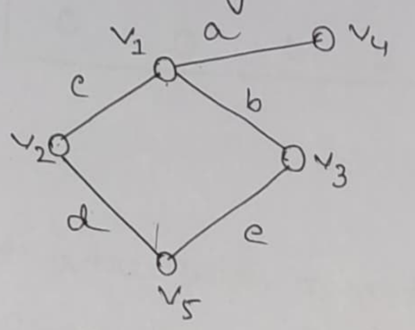
$$\sum_{odd} d(v_k) = \text{an even number} \text{ —— (iii)}$$

18

Because in eq-(iii) each $d(v_k)$ is odd. the total number of terms in the sum must be even to make the sum an even number.

e) Represent a simple graph with its equivalent adjacency and incidence matrix. — 3

n. Suppose, the simple graph is —



Adjacency matrix :-

	v_1	v_2	v_3	v_4	v_5
v_1	–	1	1	1	0
v_2	1	–	0	0	1
v_3	1	0	–	0	1
v_4	1	0	0	–	0
v_5	0	1	1	0	–

(23)

Incidence matrix

	a	b	c	d	e
v_1	1	1	1	0	0
v_2	0	0	1	1	0
v_3	0	1	0	0	1
v_4	1	0	0	0	0
v_5	0	0	0	1	1

year – 2013

2)a) prove that a tree with n vertices has $(n-1)$ edges
** Define a Binary tree. – (4+2)

Ans This theorem can be proved by induction.

Induction base :-) $e = n-1$

If $n=1$, then $e=0$

i.e., a single node no edges. or null graph that not containing any ckt. so. this is a tree.

Induction hypothesis :-) If n' be the no. of vertices and e' be the no. of edges.

$$e' = n' - 1.$$

Induction step :-) continuing this. thus.

$$n' = e' + 1$$

Now, if we add one more vertex into binary tree. then it will increase one more edge in the binary tree.

$$n'+1 = (e'+1)+1$$

Now, Considering the R.H.S

$$(e'+1)+1$$
$$= (n'-1+1)+1$$
$$= n'+1$$

$\therefore$ L.H.S = R.H.S [Proved]

Binary tree :- A binary tree is a special form of tree which contains finite set of node such that:

i) T is empty.

ii) Tree contains specially designated call the root node of T and the remaining node of T from two disjoint binary tree T_1 and T_2 which are called left sub-tree and right-sub-tree respectively.

b) What is minimal spanning tree? Explain primp's algorithm with non-trival example - (2+8)

Ans Minimal spanning tree :- A spanning tree with the smallest weight in a weighted graph is called a shortest spanning tree or shorest-distance spanning tree or minimal spanning tree.

prim's algorithm :→

Let $G = \{V, E\}$ be graph and $S = (V_S, E_S)$ be spanning tree to be found from G.

Step-1 :→ Select a vertex v_1 of V and initialize

$$V_S = \{v_1\} \text{ and } E_S = \{\}$$

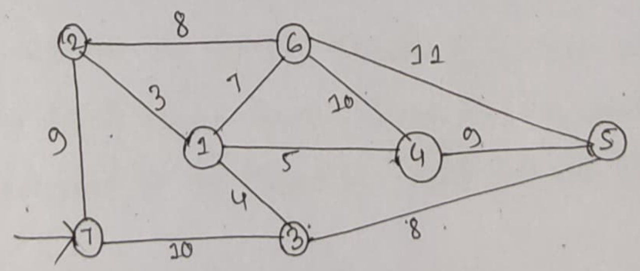
Step-2 :→ Select a nearest neighbour of v_i from V that is adjacent to same $v_j \in V_S$ and that edge (v_i, v_j) does from a cycle with members edge of E_S

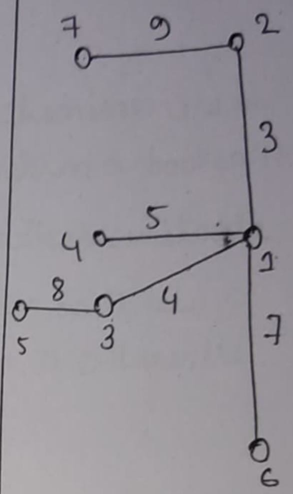
$$\text{Set, } V_S = V_S \cup \{v_i\} \text{ and}$$

$$E_S = E_S \cup \{v_i, v_j\}$$

Step-3 :→ Repeat step-2 until $|E_S| = |V| - 1$

Eg:



Starting/ choosed vertex	vertex set	Selected vertex	Spanning tree
7	(2,7) (7,3)	(2,7)	
(2,7)	(7,3) (2,6) (2,1)	(2,1)	
2,7,1	(7,3) (2,6) (1,3) (1,4) (1,6)	(1,3)	
2,7,1,3	(2,6) (1,4) (1,6) (3,5)	(1,4)	
2,7,1,3,4	(2,6) (1,6) (3,5) (4,5), (6,8)	(1,6)	
2,7,1,3,4,6	(3,5) (4,5) (6,5)	(3,5)	
2,7,1,3,4,6,5			

Total cost = 3 + 4 + 5 + 7 + 9 + 8 = 36.

a) What do you mean by graph searching? Write an algorithm for Depth first search. Suggest the modification required to convert it to Breadth First Search? (2+4+2)

Am

Algorithm for Depth First Search

To count the no. of component of a graph by using DFS.

The Depth First Search mechanism use a stack for graph traversal if maintain a boolean vector size n. If there are n-vertices available each of the vertex with in the vector will be identified by 1 if it is visited on 0 otherwise.

```
do
{
if (stack empty && not all vertex visited)
  count = count +1 // increment component
endif
} while (stack ! empty OR ! all vertex visited)
```

perform DFS traversal starting from the arbitary vertex v.

2. Repeat while (stack is not empty OR vector is mark by 1)
   begin
   if (stack is empty && ! all vertex visited)
3. Set, count = count + 1. // to count the no. of component.
4. end if
5. Choose any arbitrary vertex from the remaining vertex.
6. Repeat the above step.
7. end while
8. end procedure.

(19)

b) What is the purpose of Floyed algorithm? Explain its working principle. Derive the worst case complexity of this algorithm. (2+3+2)

Ans In many areas like transportation, cartoon motion planning, Communication, network topology design etc. problems related to finding shortest path algorithm. Floyed algorithm is one kind of shortest path algorithm.

working principle

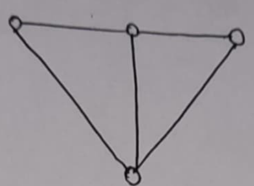
(29)

Year-2012

2) a) write an algorithm which executes a breadth-first-search ~~algorith~~ of a graph G beginning with a starting vertex A. — 6

Ans: See - Year - 2015 → 2) c)

b) Find all the spanning trees of the graph shown below



How many spanning trees can be formed? why?

— 6

(2)

(c) prove that the number of odd degree vertices in a simple graph is always even. — (4)

Ans See → 2014 → 3/(b)

3)(a) When are two graphs $G(V, E)$ and $G^*(V^*, E^*)$ are said to be isomorphic? Illustrate with a suitable example. — (6)

Ans

$f: V \rightarrow V^*$ is called graph isomorphism if

i) f is one to one and onto

ii) $a, b \in V$

$\{a, b\} \in E$

if and only if

$\{f(a), f(b)\} \in E^*$

When such a function exist G and G^* are isomorphic graph and it is written as

$$G \cong G^*$$

In other words, two graphs $G(V, E)$ and $G^*(V^*, E^*)$ are said to be isomorphic if the following four conditions are satisfied —

1. The no. of vertices in G and no. of vertices in G^* are equal.

i.e., $V = V^*$

2. The no. of edges in G and no. of edges in G^* are also equal.

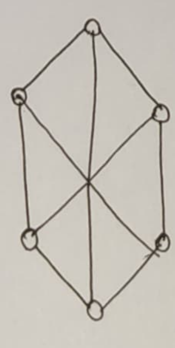
i.e., $E = E^*$

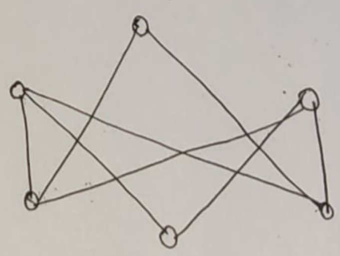
3. Both must have equal no. of vertices with same degree.

4. They must have same degree sequence and same cycle vector $(c_1, \ldots c_n)$

where c_i is no. of cycle of length i.

Eg. Consider two graphs.





G^*

The above two graphs have -

1. equal no. of vertices, i.e., 6 no. of vertices.
2. equal no. of edges, i.e., 9 no of edges.
3. They both have equal no. of vertices with degree 3.
4. They both have equal no. of degree sequence and same cycle vector.

So, the above two graphs are isomorphic as they fulfil the 4 criteria of graph Isomorphism.

(24)

3) b) Prove that, even connected graph with three or more vertices has at least two vertices which are not cut-vertices. — (4)

Ans

(8)

c) How are graphs maintained in the memory of a computer? Name two such methods and illustrate them with suitable non-trival examples. — (6)

Ans There are three ways to store a graph in memory.

1. Nodes as objects and edges as pointers.
2. A matrix containing all edge weights between numbered node x and node y.
3. A list of edges between numbered node.

The two such methods are –

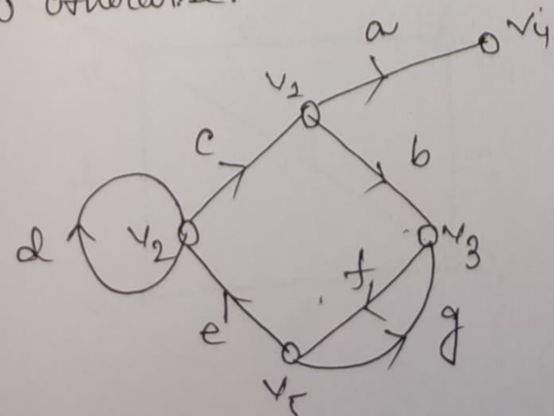
1) Adjacency matrix.

2) Incidence matrix.

1) Adjacency matrix :-

The adjacency matrix is used in representing the di-graph $x_{ij} = 1$ if there is an edge directed from ith vertex to jth. vertex $x_{ij} = 0$ otherwise.

Eg.



	v_1	v_2	v_3	v_4	v_5
v_1	0	0	1	1	0
v_2	1	1	0	0	0
v_3	0	0	0	0	1
v_4	0	0	0	0	0
v_5	0	1	1	0	0

2) Incidence matrix :→

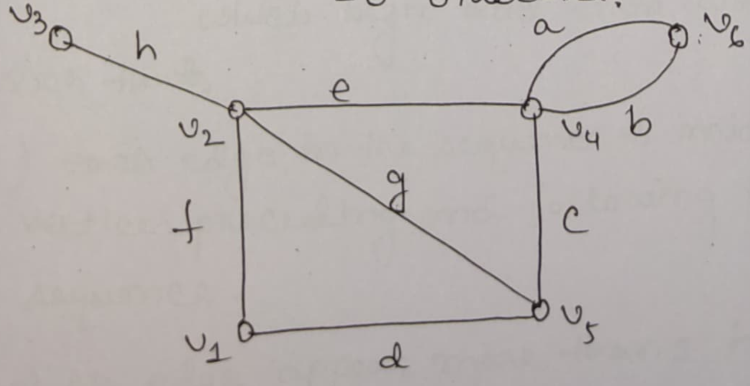
Let, G be a graph with n vertices e edges. and no self-loops. Define an n by e matrix $A = [a_{ij}]$ whose n rows correspond to the n vertices and the e columns correspond to the edges as follows.

The matrix element.

$a_{ij} = 1$, if jth edge e_j is incident on ith vertex v_i and

$= 0$ otherwise.

Eg.



	a	b	c	d	e	f	g	h
v_1	0	0	0	1	0	1	0	0
v_2	0	0	0	0	1	1	1	1
v_3	0	0	0	0	0	0	0	1
v_4	1	1	1	0	1	0	0	0
v_5	0	0	1	1	0	0	1	0
v_6	1	1	0	0	0	0	0	0

Year - 2011

2)a) Distinguish between a path and a circuit in a context of a graph. (2+2+3)

Ans A walk is define as a finite alternative sequence of vertices and edges of the form.

$(v_i, E_j)\ (v_{i+1}, E_{j+1}) \cdots (E_k, v_m)$

which begin and ends with vertices.

such that,

i) each edge in the sequence is incident on the vertices preceding and following it in the sequence.

ii) No edge appear more than 1 in the sequence is called walk or trail on G

(28)

⇒ A walk that begins and ends with same vertex is called closed walk.

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

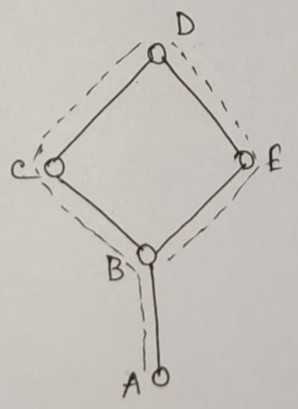
path :→ An open walk in which no vertex appear more than 1 is called path.

Circuit :→ A closed walk with atleast one edge in which no-vertex except the terminal vertex appear more than one is called circuit.

Q. What is Eular path?

Ans A path in graph G is called Eular path if it includes every edge exactly once. Since the path is called eular path or eular trail.

Eg.



---→ Signifies eular path.

prove that if a graph G has a vertex of odd-degree there can be no. eular circuit.

(29)

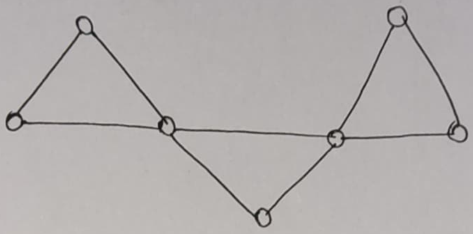
2) b) Describe the Kruskal's algorithm for finding the minimum spanning tree for a graph G. Comment about the complexity of the algorithm. (7+2)

Ans See year - 2015 → 5) c)

(30)

3.a) When is a graph said to be an arbitrarily traceable graph from ~~any~~ vertex v? Draw a graph which is not arbitrarily traceable from any of its vertices? —(2+3)

Am see → 2014 → 2) d)



b) Prove that in a complete graph G with n vertices, there are $(n-1)/2$ edge disjoint Hammiltonian circuit if n is an odd number ≥ 3. —6

Am A complete graph with n-vertices has $n(n-1)/2$ edges. and a hamiltonian circuit consists of n edges.

∴ The number of edge-disjoint hamiltonian circuits in G cannot exceed $(n-1)/2 \leq \frac{n(n-1)}{2}/n$

This implies there are $(n-1)/2$ edge-disjoint hamiltonian circuits. when n is odd it can be shown as by keeping the vertices fixed on a circle. rotate the polygonal pattern clockwise by

$$\frac{360}{n-1}, \frac{2\cdot 360}{n-1}, \frac{3\cdot 360}{n-1}, \ldots, \frac{n-3}{2}\cdot\frac{360}{n-1} \text{ degre}$$

At each rotation we get hamiltonian ckt that has no edge in common with any previous once. Thus we have $m-3/2$ new hamiltonian circuits all edges disjoint from one and also edge disjoint among themselves.

c) Define incidence matrix and adjacency matrix of a graph. Illustrate with suitable example. (2+3)

Ans See → 2012 → 3) c)

Year → 2010

2) a) Represent a simple graph with its equivalent adjacency and incidence matrix. — 4

Ans See → 2014 → 3(c)

b) Compare the breadth first search (BFS) and depth first search (DFS) graph searching. — 4

BFS	DFS
i) BFS stands for "Breadth First Search".	i) DFS stands for "Depth First Search".
ii) BFS starts traversal from the root node and then explore the search in the label by label manner. i.e., as close as possible from the root node.	ii) DFS traversal from the root node and explore the search as far as possible from the root node. i.e., depth wise.

(b)

BFS	DFS
iii) BFS can be done with the help of queue i.e., FIFO implementation.	iii) DFS can be done with the help of stack. i.e., LIFO order.
iv) This algorithm works in single stage. The visited vertices are removed from the queue and then display at once.	iv) The algorithm works in two stages - in the first stage the visited vertices are pushed onto the stack and later on when there is no vertex further to visit those are.
v) BFS is slower than DFS.	v) DFS is more faster than DFS.
vi) BFS require more memory compare to DFS.	vi) DFS require less memory compare to BFS.

c) Prove that the distance between vertices of a connected graph is a metric. - 4 * *

Ans A function $f(x,y)$ of two variables (a distance between them) is must satisfy the following criteria to become a metric.

1) Non-negativity

$$f(x,y) \geq 0 \text{ \& } f(x,y) = 0$$

if and only if $x = y$

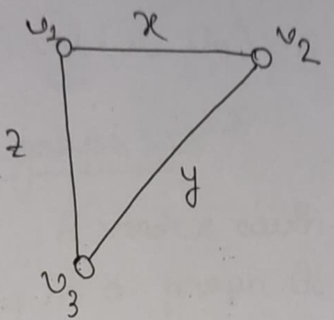
2) Symmetry $f(x,y) = f(y,x)$

(38)

3) Triangle in-equality

$f(x,y) \leqslant f(x,z) + f(z,y)$ for any z.

A function that satisfies this condition is called metric.



Now, for the above connected graph satisfies condition 1 and 2 is immediately evident. Since $d(v_1, v_3)$ is the length of shortest path between v_1 and v_3.

this path cannot longer than path between v_1 and v_3 which goes through a specified vertex v_K

$d(v_1, v_2) + d(v_2, v_3) \geqslant d(v_1, v_3)$

d) Define the terms eccentricity and center of a graph G. Illustrate with an example. 2+2

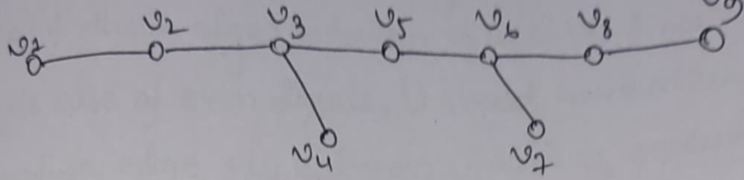
**

Ans

Essentricity :-

It also refered to as associated number on seperation of a vertex in a graph. The essentricity $E(v)$ of a vertex in a graph G is the distance from to the vertex farthest from v in G i.e., Eg: $E(v) = \max(d(v, v_i))$

Eg:

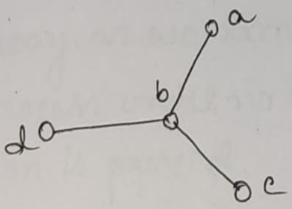


$E(v_1) = d(v_1, v_9) = 6$

<u>center of a graph:-</u> **

A vertex with minimum essentricity in a graph G is called center of G.

Eg:



for the above graph the essentricity of 4-vertices are

$E(a) = 2$

$E(b) = 1$

$E(c) = 2$

$E(d) = 2$

So, the center of graph is b.

b) Prove that a connected graph G is an Euler graph if and only if it can be decomposed into circuits. – 6

Ans Suppose graph G can be decomposed into circuits. that is, G is a union of edge-disjoint circuits. Since the degree of every vertex in a circuit is two. the degree of every vertex in G is even. Hence G is an Euler gra

(35)

Conversely, let G be an Eular graph. Consider a vertex v_1. There are at least two edges incident at v_1. Let, one of these edges between v_1 and v_2. Since vertex v_2 is also of even degree, it must have atleast another edge, say between v_2 and v_3. proceeding in this fashion, we eventually arrive at a vertex that has previously been traversed, thus forming a circuit Γ. Let us remove Γ from G. All vertices in the remaining graph (not necessarily connected) must also be of even degree. From the remaining graph remove another circuit in exactly the same way as we removed Γ from G. Continue this process until no edges are left. Hence the theorem is proved.

b) Illustrate Primp's algorithm to find the shortest spanning tree of a given graph using a suitable example. — 10

Ans See → 2013 - 2(b)

Year → 2009

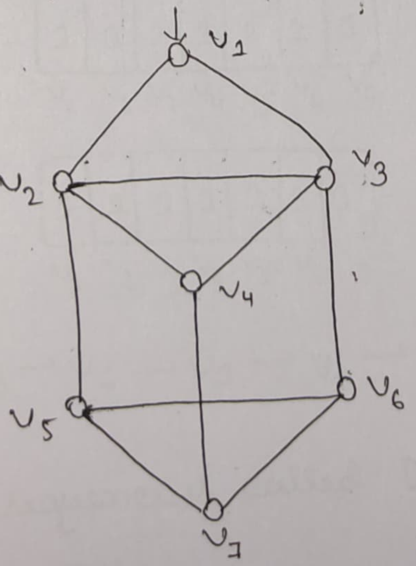
2)a) Find minimum and maximum possible height of a binary tree with n-vertices. → 4

(36)

b) Explain Depth-First-Search algorithm with a non-trival ~~eve~~ example. What modifications are to be done to get Breadth-First-Search (BFS) algorithm from it? — 5

Ans See → 2013 → 3/a) and

Eg:



Stack	v_1	v_2	v_3	v_4	v_5	v_6	v_7	Visited
v_1	0	0	0	0	0	0	0	—
v_2, v_3	1	0	0	0	0	0	0	v_1
v_2, v_4, v_6	1	0	1	0	0	0	0	v_3
v_2, v_4, v_5, v_7	1	0	1	0	0	1	0	v_6
v_2, v_4, v_5	1	0	1	0	0	1	1	v_7
v_2, v_4	1	0	1	0	1	1	1	v_5
v_2	1	0	1	1	1	1	1	v_4
	1	1	1	1	1	1	1	v_2

$v_1 \rightarrow v_3 \rightarrow v_6 \rightarrow v_7 \rightarrow v_5 \rightarrow v_4 \rightarrow v_2$

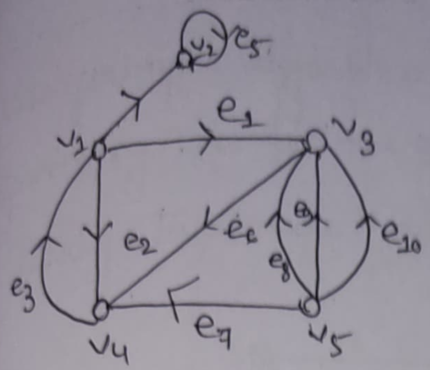
This sequence is called Depth-First-Search sequence.

c) Prove that in a simple connected graph with n-vertices ($n > 1$) at least two vertices are of equal degrees. - 3

d) What can be said about in-degrees and out-degrees of vertices of a directed graph? justify your answer - 2.

An In-degree :-) The number of edges incident into v_i is called the in-degree of v_i and is written as $d^-(v_i)$

Out degree :-) The number of edges incident out- of a vertex v_i is called out-degree of v_i and is written as $d^+(v_i)$



$d^+(v_1) = 3 \qquad d^-(v_1) = 1$

$d^+(v_2) = 1 \qquad d^-(v_2) = 2$

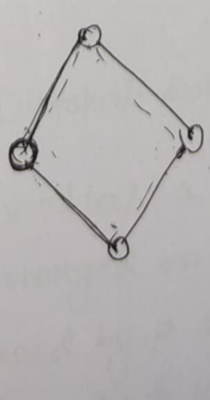
$d^+(v_5) = 4 \qquad d^-(v_5) = 0.$

3(a) prove that, if the graph G does not have any isolated vertices then it has all Eular cycle if and only if G is connected and the degree of each vertex is even. — 6

Ans Suppose P is eularian trail of G.

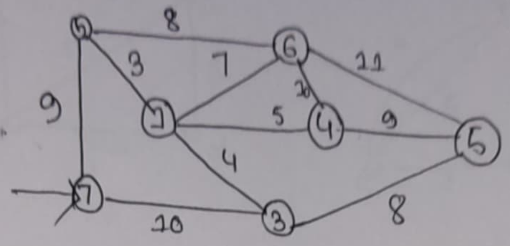
Whenever P passes through a vertex. there is a contradiction of 2 towards degree of that vertex. each edge occurs exactly once in P. each vertex must have even degree

Suppose, that the degree of each vertex is even. Since G is connected. each vertex has degree at least two. G contains a cycle C. if C contains every edge of G. then it is proved.



(40)

b) Use prim's algorithm to find the minimal spanning Tree for the graph below showing the steps clearly. Take node 7 as the starting node. — 6



Ans See → 2013 2|b)

c) Prove that a simple graph with n vertices and k components can have at most $(n-k)(n-k+1)/2$ edges — 4

Ans See → 2015 2|b)

Year → 2008

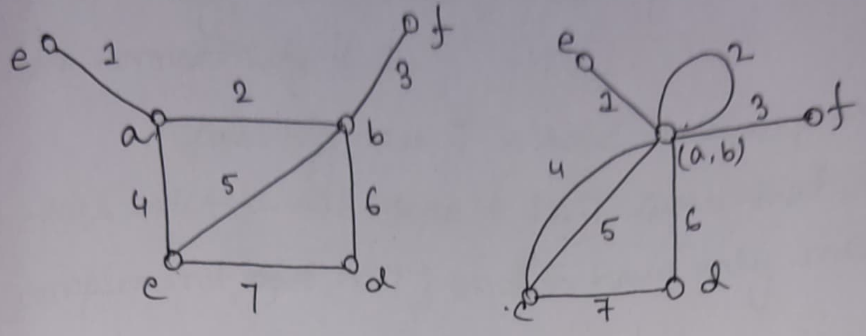
2)a) What do you mean by Arbitrarily traceable graph?

Ans Year - 2014 - 2|d)

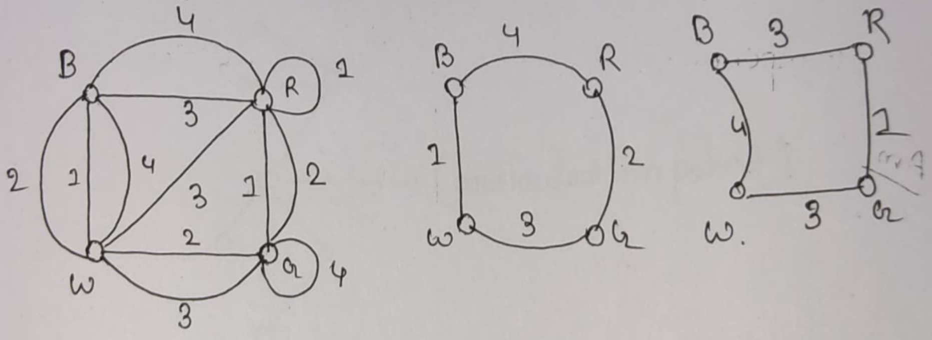
b) Define fusion, edge-Disjoint subgraph with diagram.

Ans <u>Fusion</u>:→ A pair of vertices a, b in a graph are said to be fused (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. Thus fusion of two vertices does not

(ii) alter the no. of edges but it reduces the no. of vertices by 1.



<u>Edge disjoint subset</u> :- Two subgraphs g_1 and g_2 of a graph G are said to be edge-disjoint if g_1 and g_2 do not have any edge in common.



c) Prove that a simple graph with n vertices must be connected if it has more than $[(n-1)(n-2)]/2$ edges. — 5

Am See → 2(c) → 2014

d) What is isomorphic graph?

Am See → 2012 → 2(c)

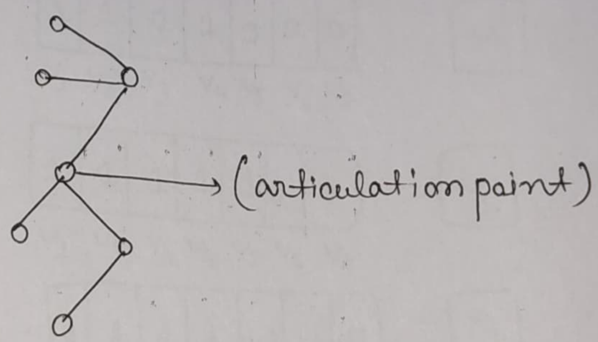
12

3) a) Explain articulation point of a spanning tree.

Ans A spanning tree is said to be seperable if its vertex connectivity is 1.

Spanning tree T is said to be seperable if there exist a sub-tree t in T. such that, $\bar{t}$ (complement of t in T) and $\bar{t}$ have only one vertex in common.

In seperable tree a vertex whose removal disconnects the tree is called cut-vertex or a cut ~~set~~ node or a articulation point.

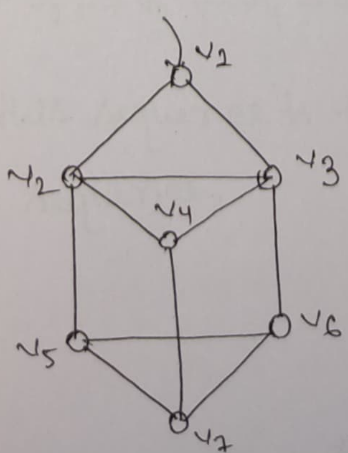


b) with an example comparatively discuss DFS and BFS graph searching method. — 8

Eg See → 2010 2 → b) and 2009 2) b) and below

for the example.

BFS



43 E

queue	vector (v_1 v_2 v_3 v_4 v_5 v_6 v_7)	vertex visited
v_1	0 0 0 0 0 0 0	—
v_2 v_3	1 0 0 0 0 0 0	v_1
v_3 v_4 v_5	1 1 0 0 0 0 0	v_2
v_4 v_5 v_6	1 1 1 0 0 0 0	v_3
v_5 v_6 v_7	1 1 1 1 0 0 0	v_4
v_6 v_7	1 1 1 1 1 0 0	v_5
v_7	1 1 1 1 1 1 0	v_6
Empty	1 1 1 1 1 1 1	v_7

$v_1 \to v_2 \to v_3 \to v_4 \to v_5 \to v_6 \to v_7$

So, this sequence is the breadth first search sequence.

(c)

c) Represent a simple graph with its equivalent adjacency and incidence matrix.

Ans See → 2014 → 3(e)

year → 2007

2)a) Write and analyse an algorithm for finding the shortest distance between two vertices of a weighted undirected graph. Illustrate your algorithm with non-trivial example.

Ans To find the shortest distance between all ~~parts~~ pairs of vertices in a weighted graph where the vertices are $v_1, v_2, \cdots v_n$ carry out following procedure.

Step-1 → For $i = 1$ to n set $d(i, i) = 0$

For $i \neq j$ if $v_i v_j$ is an edge let, $d(i, j)$ be the weight of this edge. otherwise,

set, $d(i, j) = \alpha$

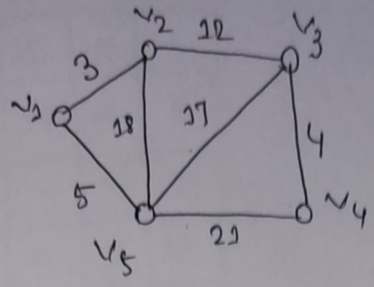
Step-2 → For $k = 1$ to n

for $i, j = 1$ to n

Let, $d(i, j) = \min\{d(i, j) \cdot d(i, k) + d(k, j)\}$

To find value of $d(i, j)$ is the shortest distance from v_i to v_j

Eg: Consider a graph G –



Iteration – 0

Source

	v1	v2	v3	v4	v5
v1	–	3	α	α	5
v2	3	–	12	α	18
v3	α	12	–	4	17
v4	α	α	4	–	21
v5	5	18	17	21	–

Destination

	v1	v2	v3	v4	v5
v1	–	2	3	4	5
v2	1	–	3	4	5
v3	1	2	–	4	5
v4	1	2	3	–	5
v5	1	2	3	4	–

Iteration – 1

Source

	v1	v2	v3	v4	v5
v1	–	3	α	α	5
v2	3	–	12	α	8
v3	α	12	–	4	17
v4	α	α	4	–	21
v5	5	8	17	21	–

	v1	v2	v3	v4	v5
v1	–	2	3	4	5
v2	1	–	3	4	1
v3	1	2	–	4	5
v4	1	2	3	–	5
v5	1	1	3	4	–

Iteration-2

	V_1	V_2	V_3	V_4	V_5
V_1	-	3	15	α	5
V_2	3	-	12	α	8
V_3	15	12	-	4	17
V_4	α	α	4	-	21
V_5	5	8	17	21	-

	V_1	V_2	V_3	V_4	V_5
V_1	-	2	2	4	5
V_2	1	-	3	4	1
V_3	2	2	-	4	5
V_4	1	2	3	-	5
V_5	1	1	3	4	-

Iteration-3

	V_1	V_2	V_3	V_4	V_5
V_1	-	3	15	19	5
V_2	3	-	12	16	8
V_3	15	12	-	4	17
V_4	19	16	4	-	21
V_5	5	8	17	21	-

	V_1	V_2	V_3	V_4	V_5
V_1	-	2	2	3	5
V_2	1	-	3	3	1
V_3	2	2	-	4	5
V_4	3	3	3	-	5
V_5	1	1	3	4	-

Iteration-4

	V_1	V_2	V_3	V_4	V_5
V_1	-	3	15	19	5
V_2	3	-	12	16	8
V_3	15	12	-	4	17
V_4	19	16	4	-	21
V_5	5	8	17	21	-

	V_1	V_2	V_3	V_4	V_5
V_1	-	2	2	3	5
V_2	1	-	3	3	1
V_3	2	2	-	4	5
V_4	3	3	3	-	5
V_5	1	1	3	4	-

(47)

Iteration - 5

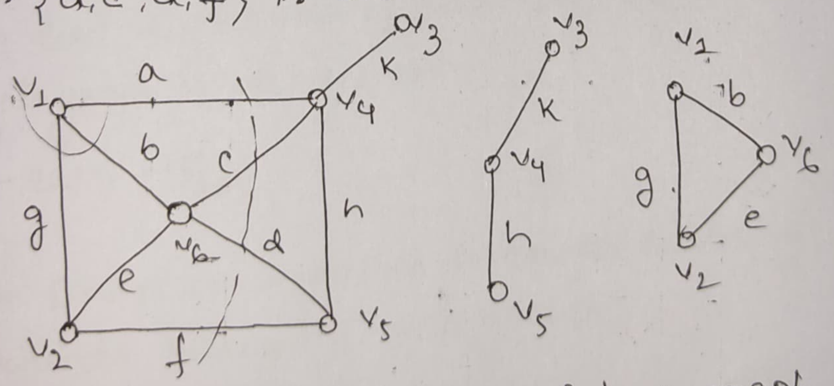
	v_1	v_2	v_3	v_4	v_5
v_1	-	3	15	19	5
v_2	3	-	12	16	8
v_3	15	12	-	4	17
v_4	19	16	4	-	21
v_5	5	8	17	21	-

	v_1	v_2	v_3	v_4	v_5
v_1	-	2	2	3	5
v_2	1	-	3	3	1
v_3	2	2	-	4	5
v_4	3	3	3	-	5
v_5	1	1	3	4	-

b) Define cut-set of a graph? What is Hamiltonian circuit?

Cut set :- In a connected graph G. a cut-set is a set of edges whose removal from G leaves G disconnected. provided removal of no proper subset of these edges disconnects G.

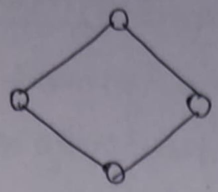
For instance in below fig. the set of edges {a, c, d, f} is a cut set.



Hamiltonial circuit :- A circuit in a graph G contains each vertex in graph exactly once. except the starting and ending vertex that appear twice is known as Hamiltonian circuit.

(18)

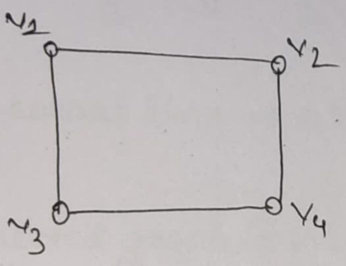
~~Ar~~



~~c) Define an algorithm to count the no. of compo~~

3)a) Define a simple connected graph.

Ans A connected graph G that not containing any self loop and parallel edges are called simple graph.

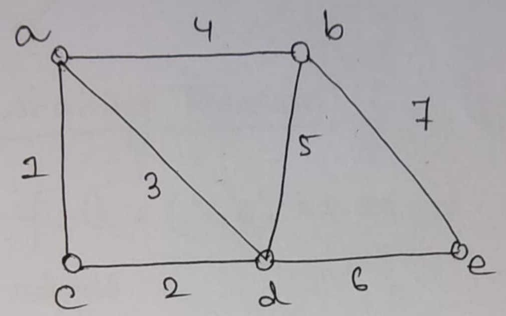


b) Prove that the number of odd-degree vertices in a simple graph is always even.

Ans See → 2014 → 3) b)

c) ~~Derive~~ Device an algorithm to count the no. of components belonging to a graph.

unicursal graph :→ In an eulor line if the initial vertex and terminal vertex are not same. we shall call such an open walk that includes all edges of a graph with-out re-tracing any edge is called open eulor line on unicursal line. a connected graph that has an uni-cursal line is called uni-cursal graph.



uni-cursal line → a 1 c 2 d 3 a 4 b 5 d 6 e 7 b

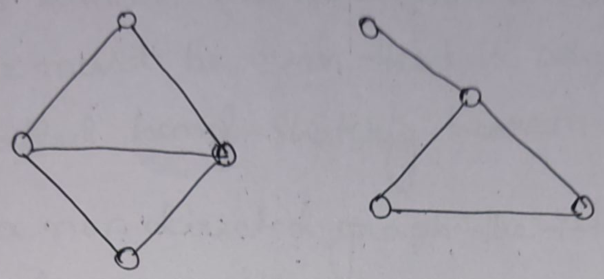
A connected graph G is called eularian if and only if the degree of each vertex of G is even.

Am Suppose p is eularian trail of G

Whenever p passes through a vertex, there is a contradiction of 2 towards degree of that vertex. each edge occurs exactly once in P. each vertex must have even degree

Suppose that. the degree of each vertex is even. since G is connected, each vertex has degree atleast two. G contains a cycle C. if C contains every edge of G. then it is proved.

[Note that, a connected graph G has an eular trail if and only if it has at-most two odd vertices]



The handshaking Theorem

if $G = (V, E)$ be an un-directed graph with e - ~~do~~ edges.

Then. $\sum_{v \in V} deg_G(v) = 2e$

i.e., the sum of degrees of the vertices in an un-directed graph is even.

proof. Since, the degree of a vertex is the number of edges incident with that vertex, the sum of the degree counts the total number of times an edge is incident with a vertex.

Since, every edge is incident with exactly two vertices, each edge gets counted twice once at each end.

Thus the sum of the degrees equal twice the number of edges.

Note :-) This theorem applies even if multiple edges and loops are present. The above theorem holds this rule that if several people shake hands. the total number of hands shake must be even that is why the theorem is called. hand-Shaking theorem.

9 In a non directed graph the total number of odd degree vertices is even.

Let, $G = (V, E)$ a non-directed graph.

proof Let. U denote the set of even degree vertices in G and W denotes the set of odd degree vertices.

Then $\sum_{v_i = V} deg_G(v_i) = \sum_{v_i = U} deg_G(v_i) + \sum_{v_i = W} deg_G(v_i)$

$\Rightarrow 2e - \sum_{v_i = U} deg_G(v_i) = \sum_{v_i = W} deg_G(v_i)$

Now, $\sum_{v_i = W} deg_G(v_i)$ is also even

$\therefore$ The no. of odd vertices in G is even

Short ans type

2025

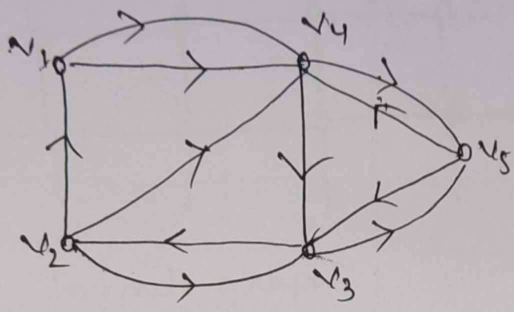
When a graph is called uni-versal graph. Give example.

6) State the konigsberg bridge problem in Graph theory.

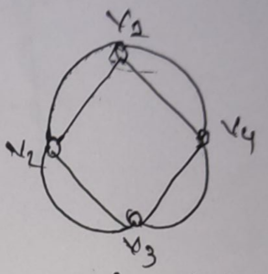
Two islands C and D formed by Pregel River in konigsberg wer connected to each other and to the banks A and B with 7 bridges as shown in fig below. The problem was to start at any of the four

What is multigraph?

In a multi-graph no- loops are allowed but more than one edge can join two vertices. ~~such~~ these edges are called multiple edges and the graph is called multi graph.



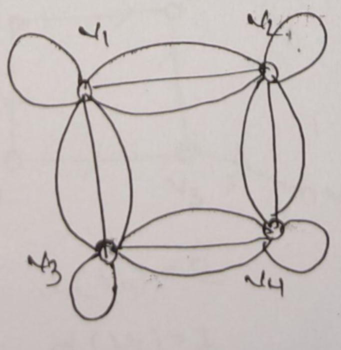
Directed multi-graph



un-directed multigraph

pseudo-graph:-)

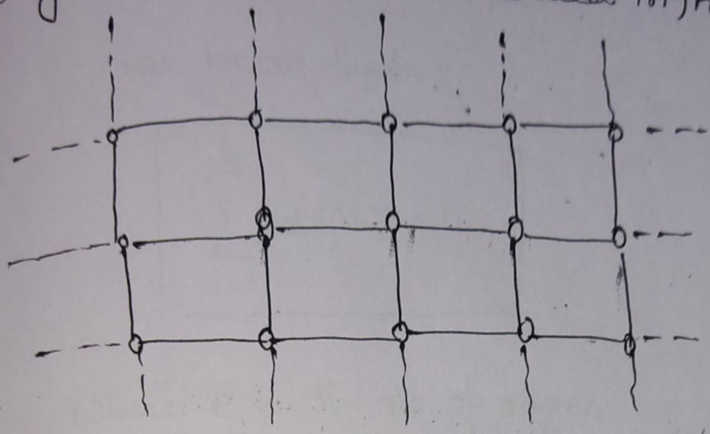
A graph in which loops and multiple edges are allowed are called pseudo graph.



pseudo graph.

Finite and Infinite graph :-

A graph with finite no. of vertices ~~and~~ as well as finite no. of edges is called finite graph. Otherwise it is called infinite graph.

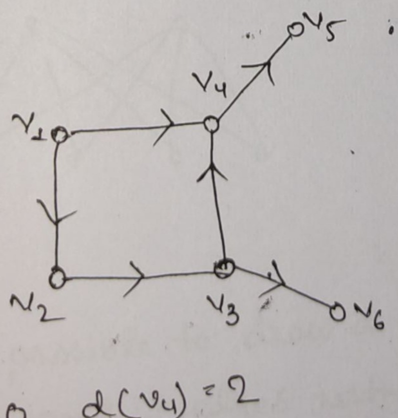


Infinite graph.

Degree of a vertex :-

The no. of edges incident on a vertex v_i with the self loops counted twice is called degree of (v_i) and is denoted as:

$d(v_i)$ or $\deg_G(v_i)$



$d(v_1) = 0$ $d(v_4) = 2$

$d(v_2) = 1$ $d(v_5) = 1$

$d(v_3) = 1$ $d(v_6) = 1$

Problem 1

1) Determine the no. of edges in a graph with 6 vertices, 2 of degree 4 and 4 of degree 2. Draw two such graph.

We know that,

$$\boxed{\sum_{i=1}^{6} d(v_i) = 2e}$$

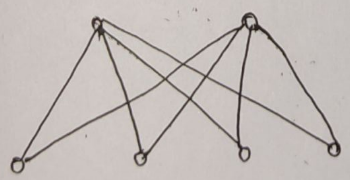
where e is the no. of edges

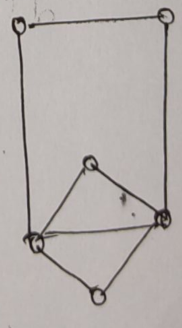
$$d(v_1) + d(v_2) + d(v_3) + d(v_4) + d(v_5) + d(v_6) = 2e$$

$\Rightarrow 4+4+2+2+2+2 = 2e$

$\Rightarrow 16 = 2e$

$\Rightarrow \boxed{e = 8} \leftarrow$ Required no. of edges -





2) It is possible to draw a graph with 4 vertices and 7 edges justify.

A simple graph with n vertices the maximum no. of edges will be - $\frac{n(n-1)}{2}$

∴ A simple graph with 4 vertices can have atmost

$\frac{4(4-1)}{2} = \frac{4 \times 3}{2} = 6$ edges.

∴ A simple graph with 4 vertices cannot have 7 edges.

∴ such a graph does not exist.

* This graph is complete graph. where each vertex is connected with every other vertex adjacently or dedicatedly. No self loops or parallel edges are allowed.

Theorem

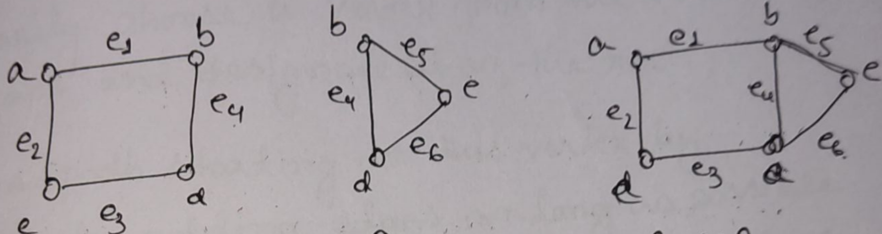
Show that for a simple graph (undirected) of n vertices the maximum no of edges $= \frac{n(n-1)}{2}$

Operational graph

Union:- Given two graph G_1 and G_2. Their union will be a graph such that,

$$V(G_1 \cup G_2) = V(G_1) \cup V(G_2)$$

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

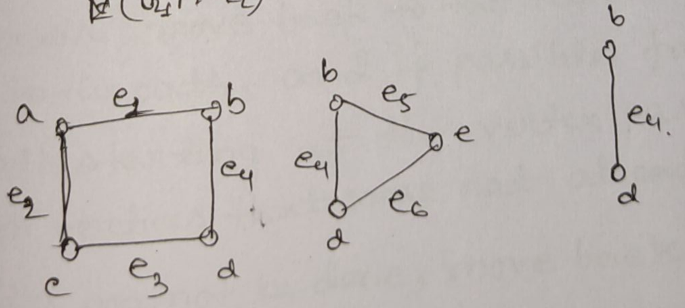


G_1 $\qquad$ G_2 $\qquad$ $G_1 \cup G_2$

Intersection:- Given two graph G_1 and G_2 such that, their intersection will be a graph

$$V(G_1 \cap G_2) = V(G_1) \cap V(G_2)$$

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



D.F.S algorithm

D.F.S search which proceeds to successive levels in a tree at the earliest possible oppartunity.

D.F.S is also called back tracking

procedure

I. Arbitarily choose a vertex from the vertices of the graph. and designate it as the root.

II. Form a path starting at this vertex by successively adding edges as long as possible where each new edge is incident with the last vertex in the path. without producing any cycle.

iii. If the path goes through all vertices of the graph. the tree consisting of this path is a spanning tree.

otherwise, move back to the next to last vertex in the path, and if possible, from a new path starting at this vertex passing through vertices that were not already visited

IV. If this cannot be done, move back another vertex in the path. that is, two vertices back in the path and repeat.

V. Repeat this procedure, begining at the last vertex visited, moving back up the path one vertex at a time. forming new paths that are as long as possible until no more edges can be added.

[vi] This process ends since the graph has a finite number of edges and is connected. A spanning tree is produced.

$$E = I + 2n$$

Induction base:-

if $n = 1$.

then $I = 0$, that is no internal path

and $E = 2$, two external path.

Induction hypothesis:-

Now consider, for $E' = I' + 2(n-1)$. —— (1)

for $(n-1)$ nodes where $n \geq 2$.

Induction ~~hypothesis~~ steps:-

Let, a tree T with n nodes and choose a leaf node at depth d.

Remove it and make it I' with E' external path and I' internal path.

Internal path will be reduced by 1 at depth d.

$$I' = (I - d)$$

External path will reduced by 2 at depth $(d+1)$ and also add an extra node at depth d. (Removal of leaf node)

$$E' = E - 2(d+1) + d$$
$$= E - 2d - 2 + d$$
$$= E - d - 2.$$

Now put E' and I' at eq^n (i)

$$E - d - 2 = (I - d) + 2(n-1)$$
$$\Rightarrow E = \cancel{d} + \cancel{2} + I - \cancel{d} + 2n - \cancel{2}$$
$$\Rightarrow E = I + 2n \text{ [Proved]}$$