Graph Theory:-

* $G = (V, E)$. It is collection of vertices and edges

$V = \{v_1, v_2, v_3, \_\_\_\_\_\_, v_n\}$

$E = \{e_1, e_2, e_3, \_\_\_\_\_\_, v_n\}$

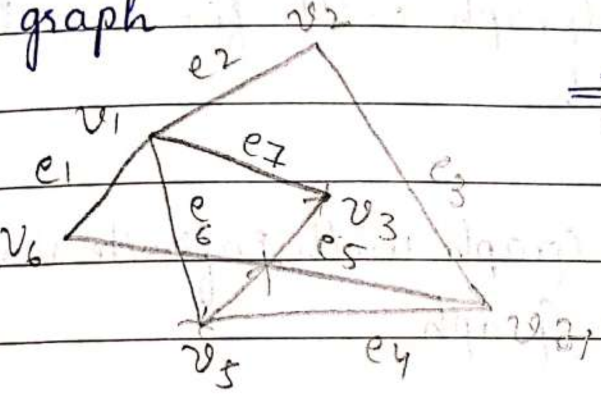
n - finite number

V - vertices

E - edges

G - graph

Ex:-



⇒ (v_1, v_6) represents edge 1.

Types of graph:-

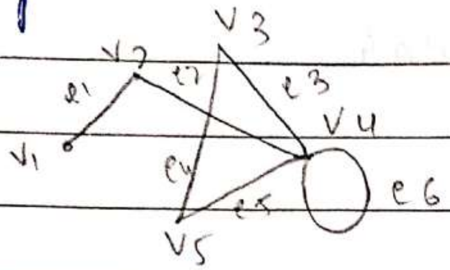
1) Finite Graph: Graph in which there are finite no. of vertices and edge

2) Infinite Graph: Graph in which there are infinite no. of vertices and edge

A graph $G = (V, E)$ consists of V non empty set of vertices (or points or nodes) and E a set of edges (or lines) Each edge has either one or two vertices associated with it called its end points

Ex:- Suppose a network is made up of datacentres and communication links between computers. We can represent the location of each datacentre by a point and each communication link by edge or line segment

Ex:

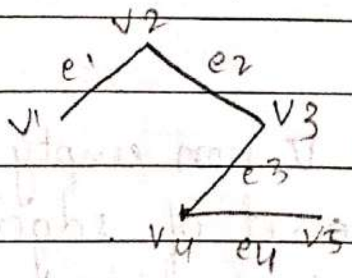


Finite Graph: The Graph with finite vertex set is called a finite graph.

Infinite Graph: The Graph with infinite vertex set is called an infinite Graph.

A Simple Graph: A graph in which each edge connects two different vertices where no two edges connect the same pair of vertice is called a simple graph.

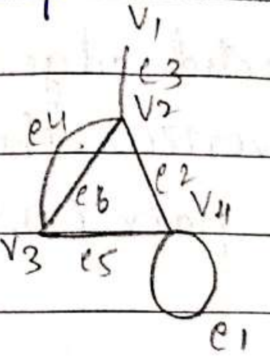
Ex:



MultiGraph: The Graphs that may have multiple edges connecting the same pair of vertices are known as multigraphs.

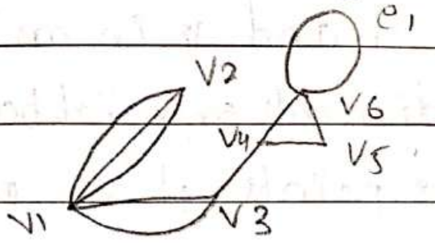
* Loops: An edge that connects a vertex to itself is called a loop

Ex:



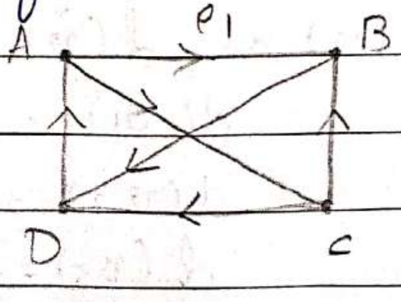
Edge e_1 is a loop

* Pseudograph: A graph that may include loops and multiple edges connecting same pair of vertices is known as pseudograph



* Round Robin tournament: A tournament where each team plays each other team exactly once is called Round Robin tournaments. Such tournaments can be modelled using directed graphs where each team is represented by a vertex. Note that, (A, B) is an edge if team A beats team B

Ex:-



(A, B), (A, C)
(C, D), (C, B)
(B, D), (D, A)

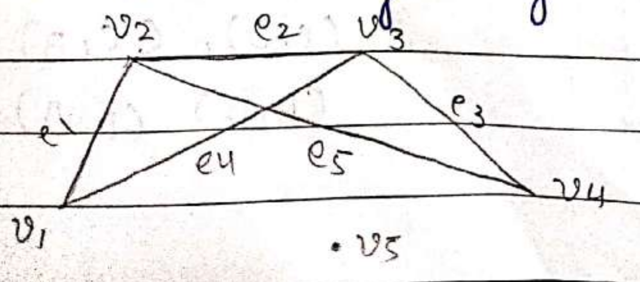
Directed or Di Graphs: A directed graph $G=\{V,E\}$ consists a non-empty set vertices V and a set of directed edges E. Each directed edge is associated with an ordered pair of vertices. The directed edge associated with ordered pair (u,v) is said to start at u and end at v.

Mixed Graph: A graph with both directed and undirected edges

Adjacency: Two vertices u and v in an undirected graph G are called adjacent or neighbours in G if u and v are the end points of an edge e

If e is associated with $\{u,v\}$ the edge e is said to be incident with the vertices $\{u,v\}$

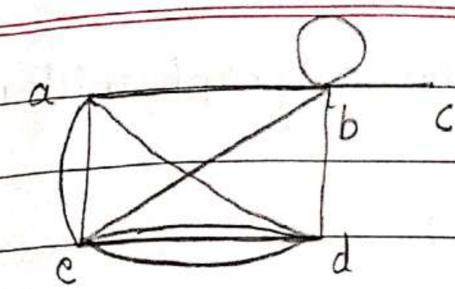
Degree of vertex in an undirected graph is the number of edges incident with it except that a loop at a vertex contributes twice to the degree of that vertex. The degree of vertex v is denoted by $\deg(v)$ or $d(v)$



$d(v_1)=2$ $d(v_4)=2$
$d(v_2)=3$ $d(v_5)=0$
$d(v_3)=3$

classmate
Date
Page

Ex:-

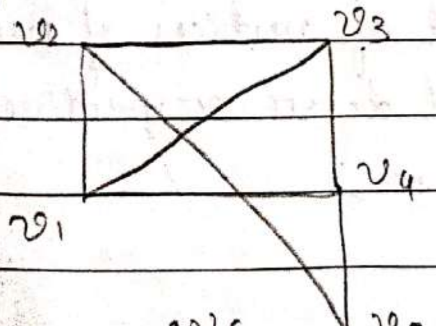


$d(a) = 4$ $d(d) = 5$
$d(b) = 6$
$d(c) = 1$
$d(e) = 6$

Isolated vertex is a vertex of degree 0 (no edges are connected to it)
Pendent vertex is a vertex is pendent if and only if it has degree one.

The Handshaking theorem: If $G = (V, E)$ be an undirected graph with 'e' edges then $\sum d(u) = 2e$ where $u \in V$
Let G be the graph i.e. $G = (V, E)$ each edge contributes two to a sum of degrees of the vertices ∵ an edge is incident with exactly two vertices ∴ the sum of the degrees of vertices is twice the no. of edges.

Ex:-



$deg(v_1) = 3$ $deg(v_4) = 3$
$deg(v_2) = 3$ $deg(v_5) = 2$
$deg(v_3) = 3$ $deg(v_6) = 0$
$e = 7$

$$\sum_{u \in V} deg(u) = 2e$$

$14 = 2(7)$
$14 = 14$

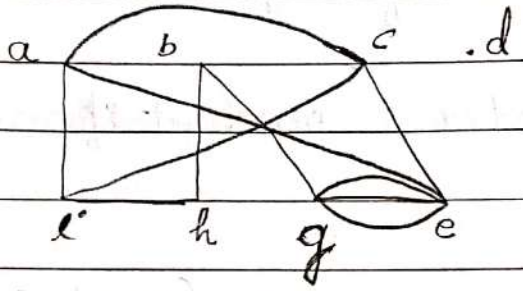
Ex:- How many edges are there in a graph with 10 vertices each of degree 6

$$\sum_{u \in V} \deg(u) = 2(e)$$

$$6(10) = 2(x)$$

$$x = 30$$

I.P. 3:-



$d(a) = 3$

$d(b) = 2$

$d(c) = 3$

$d(d) = 0$

$d(e) = 5$

$d(i) = 3$

$d(h) = 2$

$d(g) = 4$

$$\sum d(u) = 2(e)$$

$$22 = 2(x)$$

$$x = 11$$

Prove that an undirected graph has an even number of vertices of odd degree.

Let v_1 and v_2 be the set of vertices of even degree and set of vertices of odd degree respectively.

By handshaking theorem,

$$\sum_{u \in V} \deg(u) = 2e$$

$$\sum_{u \in v_1} \deg(u) + \sum_{u \in v_2} \deg(u) = 2e$$

Scanned by CamScanner

$$\sum_{u \in V_1} deg(u) + \sum_{u \in V_2} deg(u) = even.$$

$$even + \sum_{u \in V_2} deg(u) = even$$

$$\sum_{u \in V_2} deg(u) = even - even = even$$

Sum of odd degree vertices is even
∴ There are even number of odd vertices

*** Some special simple graphs:-

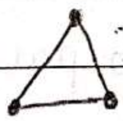
1. Complete Graphs: (K_n)
The complete graph on n vertices denoted by K_n is a simple graph that contains exactly one edge between each pair of distinct vertices

K_1 : ·

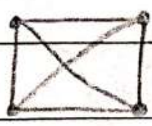
1-0

2-1 K_2 : 

K_7:

3-3 K_3: 

4-6

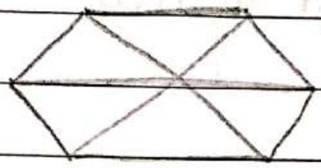
K_4: 

7-21

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

Degree of each vertex = $n-1$

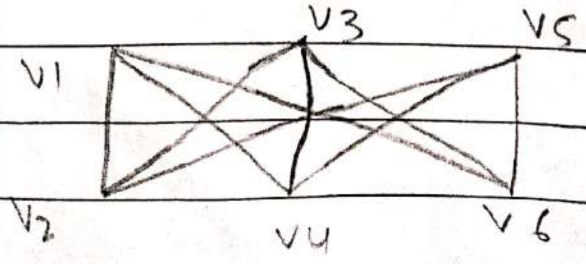
2) Regular Graph: A graph G is said to be R-regular if degree of each vertex is 'R'

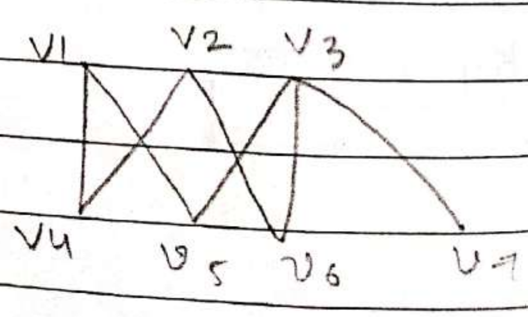


* All regular graphs are not complete
* All complete graphs are regular

3 - Regular

3) Bipartite graph: Bipartate graph is a simple graph is said to be bipartate if its vertex set V can be partioned into 2 disjoined sets V_1 and V_2 such that no edge in G connects inner two vertices in V_1 and 2 vertices in V_2. There may be an edge between one set of vertices and another set of vertices.
Here, the pair V_1, V_2 is called bipartition of the vertex V in G.



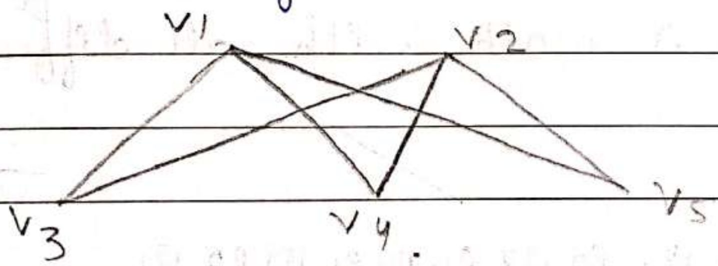


4) Complete Bipartate Graph: – ($K_{m,n}$) is the graph that has its vertex set partioned into two subsets of m, n vertices respectively

There is an edge b/w two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset and every vertex of one set is adjacent to every vertex of second set.

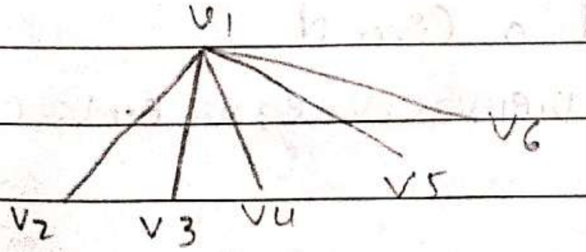
No. of vertices = m+n

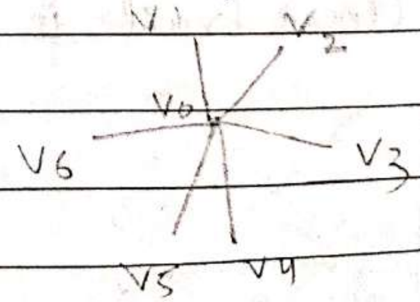
edges = m·n



5) Star Graph: Complete bipartate graph $K_{m,n}$ where m = 1 or $K_{1,n}$ is known as star graph.

Eg:-





No. of vertices = n+1

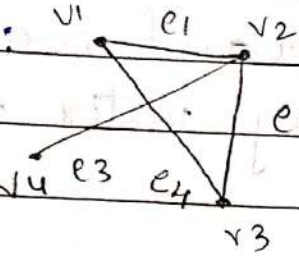
No. of edges = n

Walk:- A walk of length k from vertex v_1 to v_k is a non-empty graph (G equal to (V, E)) $G = (V, E)$ of the form $W = v_1 e_1 v_2 e_2 \ldots e_k v_k$

-OR-

walk is a sequence of vertices and edges

G:



$W_1: v_1 e_1 v_2 e_2 v_3 e_4 v_1$

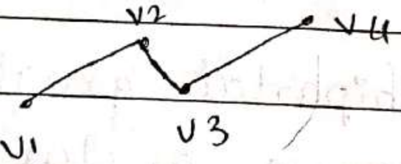
$W_2: v_2 e_2 v_3 e_4 v_1 e_1 v_2 e_3 v_4$

Trail: A trail is walk with all different edges

Path: A path is walk with all different vertices and edges

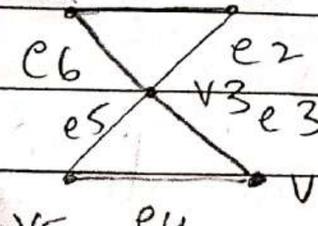
Ex: Trail: $W_2: v_2 e_2 v_3 e_4 v_1 e_1 v_2 e_3 v_4$

Ex. Path:



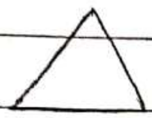
* Closed trail is called a Circuit

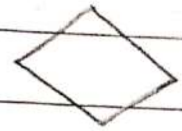
Ex.

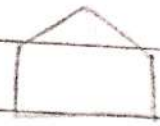


$C: v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_3 e_6 v_1$

Cycle: A closed path is called Cycle. It is denoted by C_n where $n \geq 3$

C_3: 

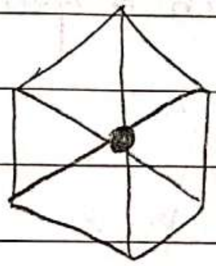
C_4: 

C_5: 

No of vertices = no. of edges = n

Wheel Graph: We obtain a wheel graph W_n when we have an additional vertex to the cycle and connect this new vertex to each of the remaining vertex by new edges.

W_n

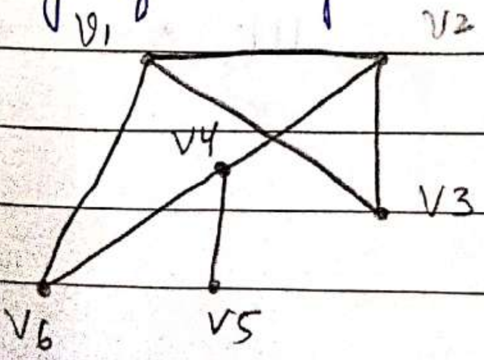


vertices n

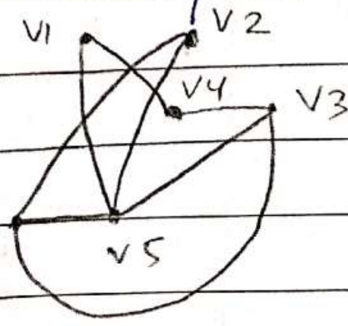
when no. of edges = $2n-2$

Complementary Graph: Suppose there is a graph G
The complementary graph $\bar{G}$ of simple graph G has the same number of vertices as G and the two vertices in $\bar{G}$ are adjacent if and only if they are non-adjacent in G

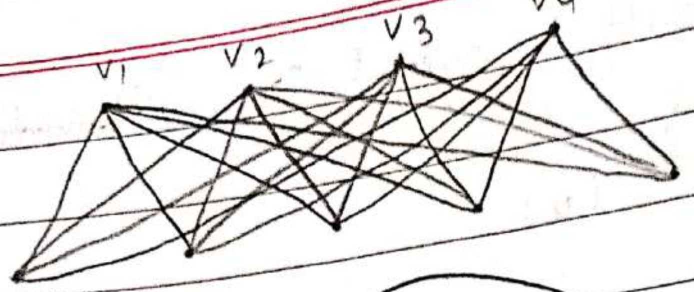
G:



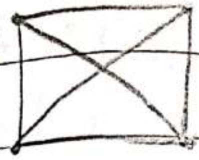
$\bar{G}$:



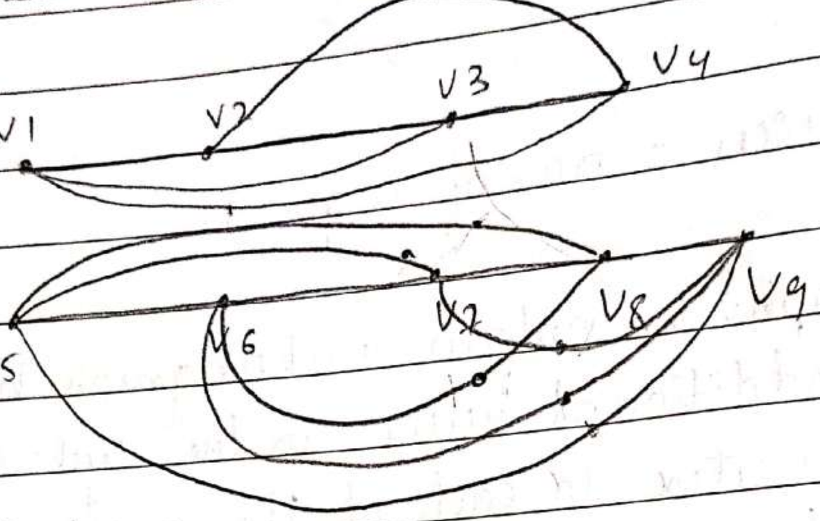
K_4:



K_4:

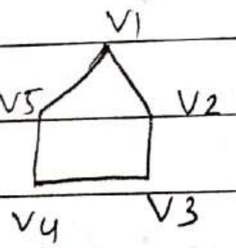


$\bar{K_4}$:

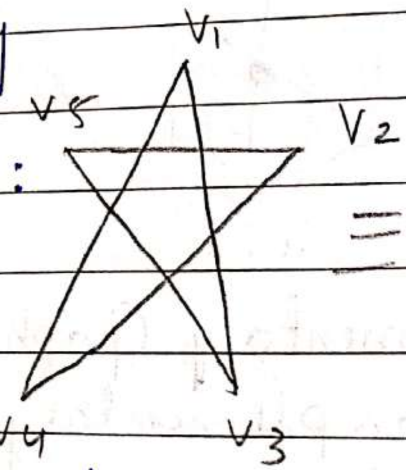


Self-complementary Graphs: If complement graph $\bar{G}$ of a graph G is itself then such a graph is called as self complementary

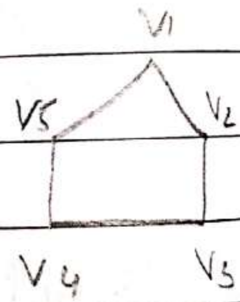
C_5



$\bar{C_5}$:

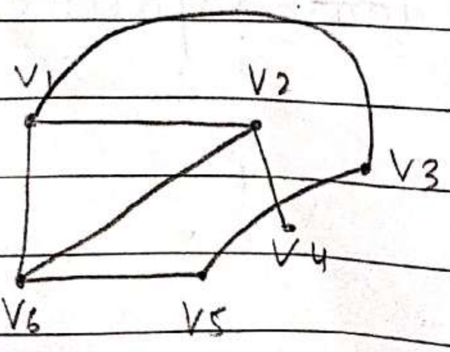


=

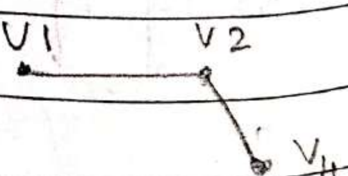


Sub-Graph: Sub-graph of a graph $G = (V, E)$ is a graph $H = (U, W)$ if $U \subseteq V$ and $W \subseteq E$

G:



H1:

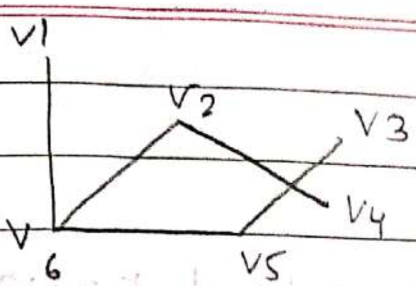


$G = (V, E)$

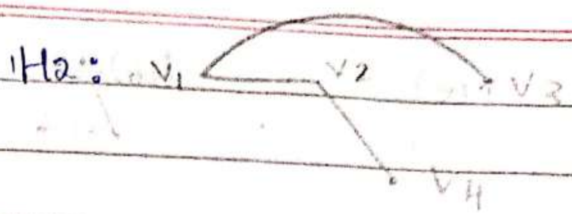
$H = (U, W)$

$U \subseteq V$, $W \subseteq E$

H_3:

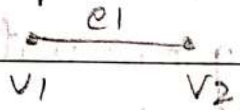


H_2:



* How many sub-graphs with atleast one vertex does the following graph have.

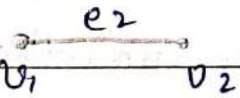
(I) K_2



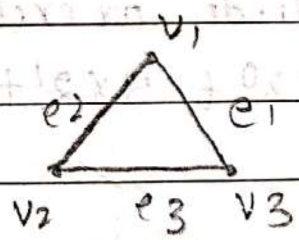
$K_2 = \underline{\underline{4}}$

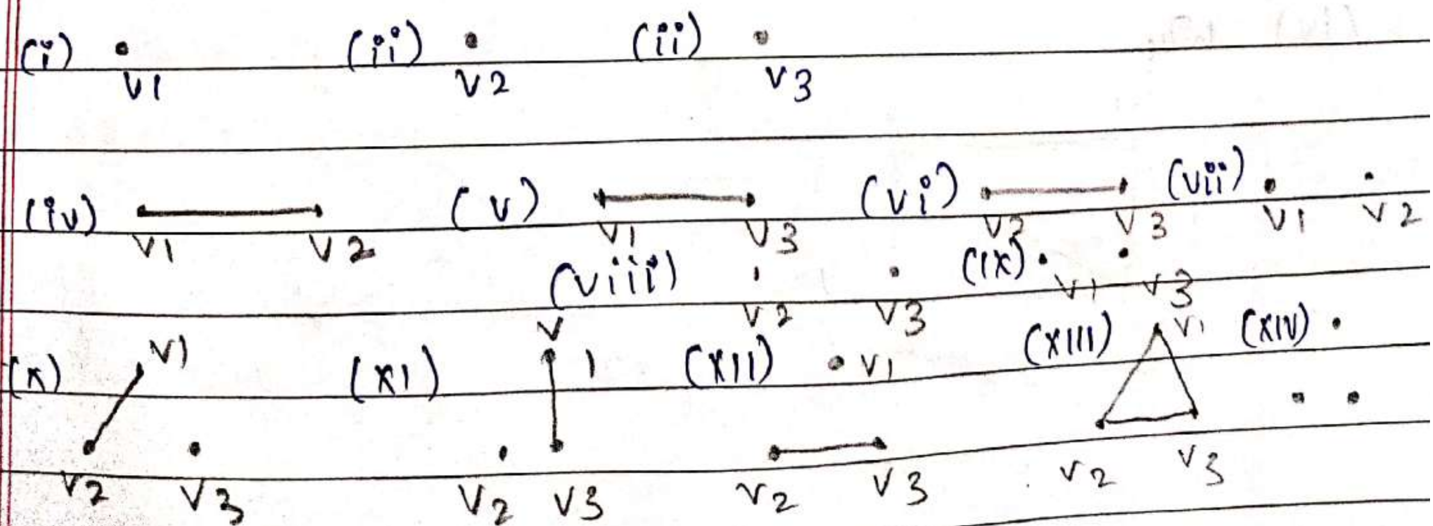
(i) $\cdot v_1$ (ii) v_2

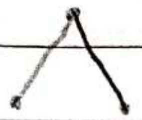
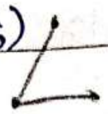
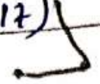
(iii) v_1 v_2 (iv)



(II) K_3 :





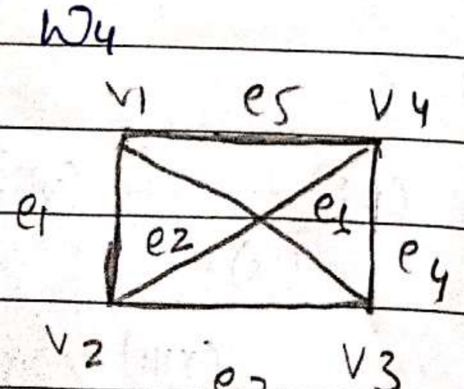
(15)  (16)  (17) 

No. of subgraphs of $K_3 = 17$

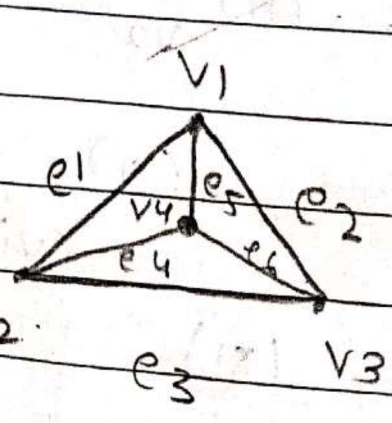
// Only for complete Graph
Selection of vertices can be done in
$3C_1$ ways = 3 (Selecting one vertex out of three vertices)
$3C_2$ ways = 3 (Selecting two vertices out of three vertices
$3C_3$ ways = 1 (Selecting three vertices out of three vertices
that can have atmost 0, 1, 3 edges respectively
As we know that the no. of subsets of a set with n elements is 2^n ∴ there are
$2^0, 2^1, 2^3$ ways to include or exclude edges
∴ No. of subgraphs = $3 \times 2^0 + 3 \times 2^1 + 1 \times 2^3 = 17$

(iii) $W_3 = K_3$

(iv) W_4



≡



$4C_1 = \frac{4!}{3! \times 1!} = \frac{4 \times 3!}{3!} = 4$ // 0 edges

$4C_2 = \frac{4!}{2! \times 2!} = \frac{\cancel{4}^{2} \times 3 \times \cancel{2!}}{\cancel{2} \times \cancel{2!}} = 6$ // 1 edges

$4C_3 = \frac{4!}{1! \times 3!} = \frac{4 \times \cancel{3!}}{\cancel{3!}} = 4$ // 3 edges

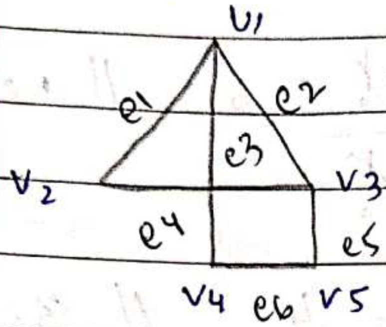
$4C_4 = 1$ // 6 edges.

Total no. of subgraphs $= 4 \times 2^0 + 6 \times 2^1 + 4 \times 2^3 + 1 \times 2^6$
$= 4 + 12 + 32 + 64$
$= 16 + 32 + 64$
$= 112$ subgraphs

* Adjacency Matrix :- Suppose that $G = (V, E)$ is a simple Graph with $|V| = n$ (// n vertices) suppose that the vertices of G are listed arbitrarily as $1, 2, \ldots n$. The adjacency matrix A or A_G is a non-zero matrix with 1 as its ij^{th} entry if i is adjacent to j or ith vertex is adjacent to jth vertex otherwise zero.

$$a_{ij} = \begin{cases} 1, & \text{if } i\text{th vertex is adjacent to } j\text{th vertex} \\ 0, & \text{otherwise} \end{cases}$$

Ex.



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

* Incidence Matrix : Let $G = (V, E)$ be an undirected graph. Suppose that $1, 2 \ldots n$ are the vertices and $E_1, E_2 \ldots E_m$ are the edges of G then the incidence matrix w.r.t. this ordering $[(V, E)]$ of V and E is I or M

$$= [m_{ij}] = \begin{cases} 1, & \text{if } e_j \text{ is incident with } i\text{th vertex} \\ 0, & \text{otherwise} \end{cases}$$

M = //3

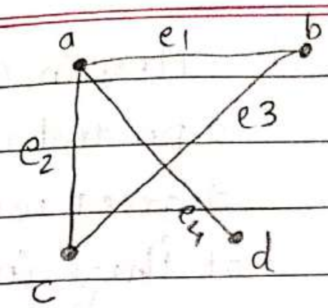
degree ←

	e_1	e_2	e_3	e_4	e_5	e_6
v_1	1	1	1	0	0	0
v_2	1	0	0	1	0	0
v_3	0	1	0	1	1	0
v_4	0	0	1	0	0	1
v_5	0	0	0	0	1	1

↓ 2

1.P11:-

(i)



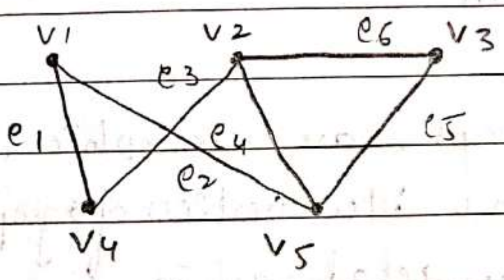
Adjacency:

	a	b	c	d
a	0	1	1	1
b	1	0	1	0
c	1	1	0	0
d	1	0	0	0

Incident:

	e1	e2	e3	e4
a	1	1	0	1
b	1	0	1	0
c	0	1	1	0
d	0	0	0	1

(ii)



Adjacency:

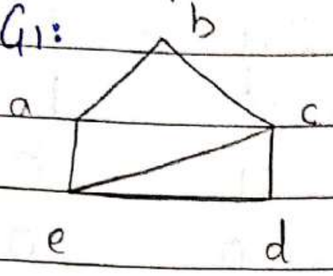
	v1	v2	v3	v4	v5
v1	0	0	0	1	1
v2	0	0	1	1	1
v3	0	1	0	0	1
v4	1	1	0	0	0
v5	1	1	1	0	0

Incident

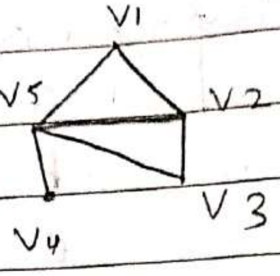
	e1	e2	e3	e4	e5	e6
v1	1	0	1	0	0	0
v2	0	0	1	1	0	1
v3	0	0	0	0	1	1
v4	1	1	0	0	0	0
v5	0	1	0	1	[illegible]	0

Isomorphism :-

LP14:- Ex: G_1:



G_2:

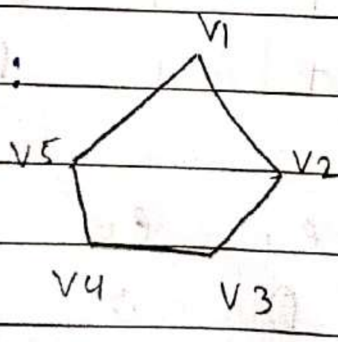


Here in G_2 we ha a pendent vertex wi degree 1 which i not there in G_1 so $G_1 \ncong G_2$

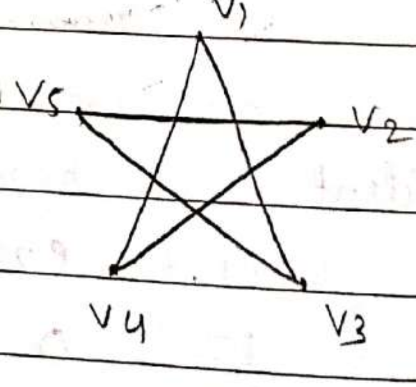
The Simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is one to one and onto function f from V_1 to V_2 with the property that a, b are adjacent in G_1 iff $f(a)$ and $f(b)$ are adjacent in G_2 for all a and b in V_1. Such a function f is called an isomorphism.

When two simple graphs are isomorphic there is one to one correspondence b/w vertices of graph that preserves the adjacency relationship

Ex: C_5:



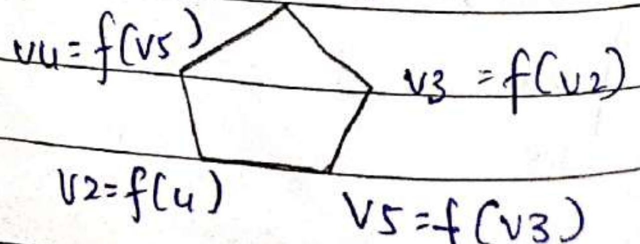
$\overline{C_5}$:



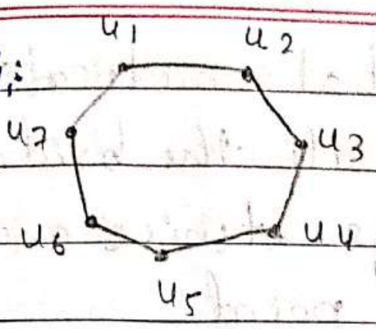
OR

$\therefore C_5 \cong \overline{C_5}$

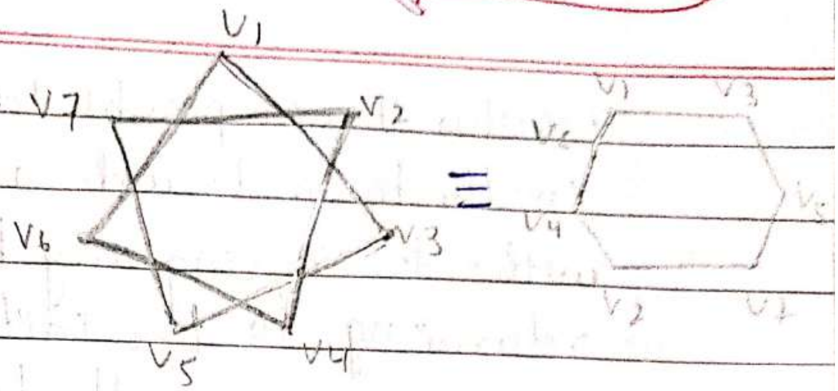
|||



IP15: G_1:



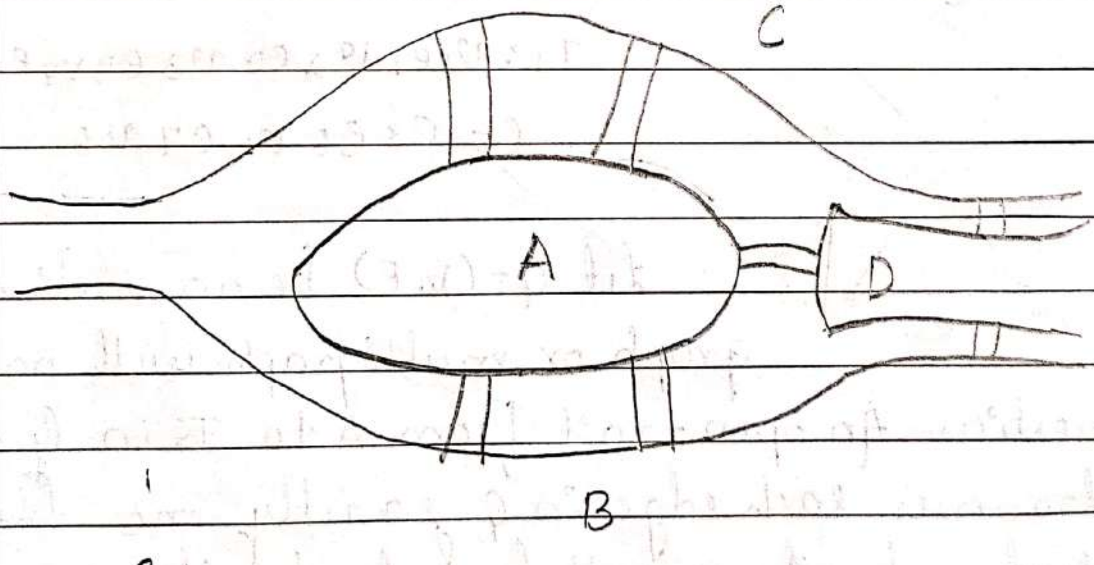
G_2:

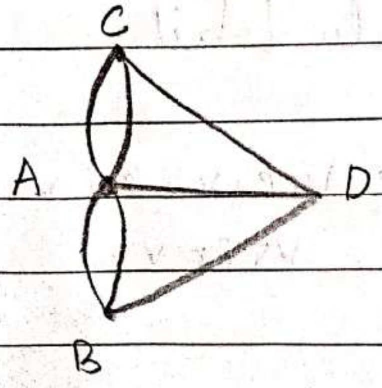


$f(v_1) = v_1$
$f(v_2) = v_3$
$f(v_3) = v_5$
$f(v_4 = v_7$

$f(u_5) = v_2$
$f(v_6) = v_4$
$f(v_7) = v_6$

* Konigsberg bridge problem:- (graph theory creator)





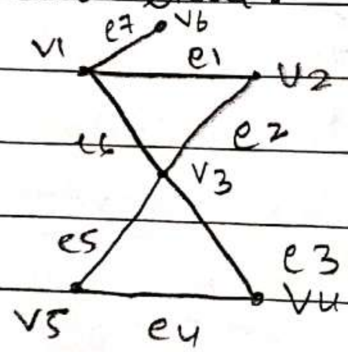
The town of Konigsberg was divided into 4 section by the branches of the pregel river. In the 18th century 7 bridges connected these regions as show in the fig spanning subgraph: all vertices

Whether it was possible to start at some location in the town, travel across all the bridges without crossing any bridge twice and returning to the initial point

The multigraph representing the bridges has 4 vertices of odd degree it does not have an Euler circuit. There is no way to start at given pt cross each bridge exactly once and return to the starting point.

Euler trail:

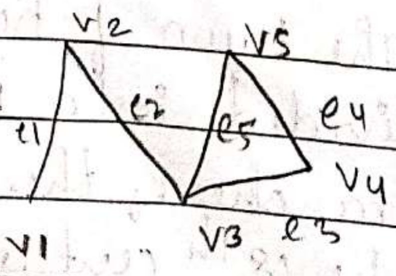
G:



T_1: $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_3 e_6 v_1 e_7 v_6$.

Let $G=(V,E)$ be an undirected graph or multigraph with no isolated vertices. An open trail from A to B in G and this traverses each edge in G exactly once then such a trail is called Euler trail.

G1:

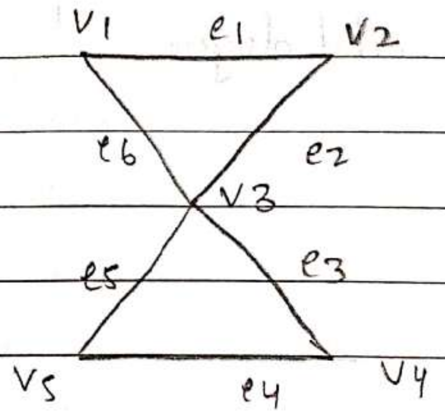


T_2: $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_3$.

Scanned by CamScanner

classmate
Date
Page

* Euler Circuit: Let $G=(V,E)$ be an undirected graph with no isolated vertices then G is said to have an Euler Circuit. If there is a circuit in G that traverses every edge of the graph exactly once.



$v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_3 e_6 v_1$

Theorem 1:-

A connected multi-graph with atleast two vertices has an Euler Circuit if and only if each of its vertices has even degree

Theorem 2:-

A connected multi-graph has an Euler trail but not an Euler Circuit iff it has exactly two vertices of odd degree

E.P 19) (iii) There are exactly 2 vertices with degree 3.
Hence it is Euler trail

a c d e b d a b

(iv) bcdbad, Euler trail
∴ Exactly two vertices with odd degree.

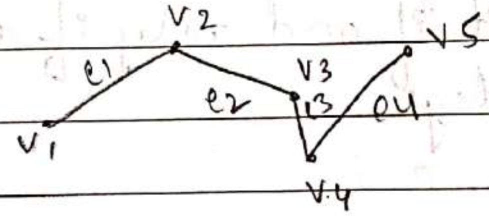
(v) Euler trail
bcdefgabgcfd
Exactly two vertices with odd degree

(vi) Neither circuit nor trail
(ii) Neither circuit nor trail

(i) Euler Circuit.

Hamilton Path

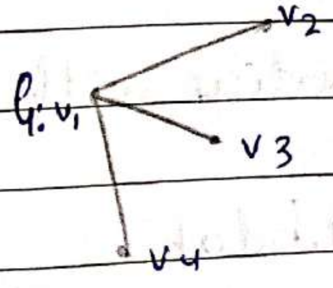
P_4:



P: $v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5$

P_4 is Hamilton Graph

A simple path in a graph G that passes through every vertex of G exactly once is called a Hamilton Path.

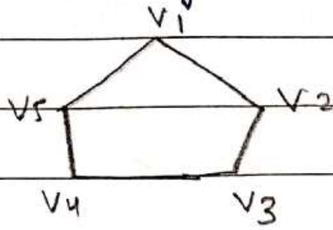


Here G is not Hamilton Graph

Hamilton Cycle:

A cycle in graph G that passes through every vertex exactly once is called Hamilton Cycle

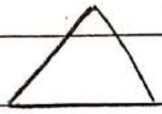
C_5:



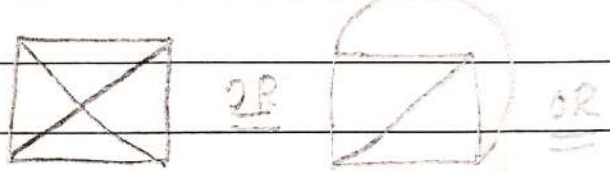
C_5 is Hamilton cycle.

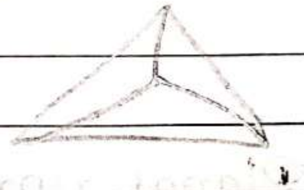
Planar Graph:

K_3:

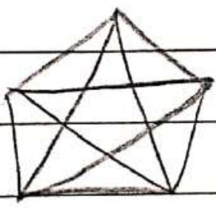


K_4:





K_5:

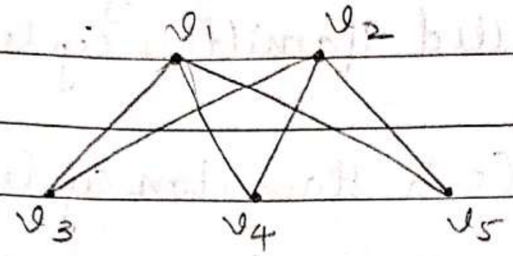


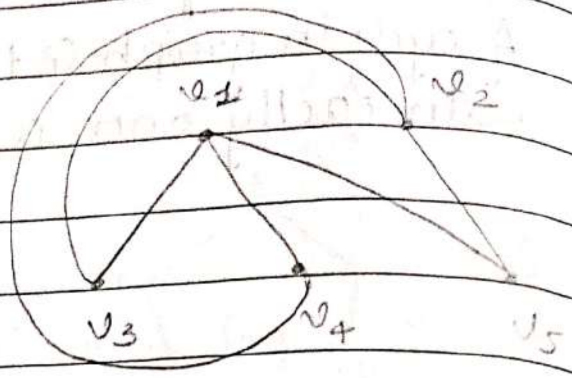
K(3,3) planar

A graph is said to be planar graph if it can be represented by a figure drawn on plane such a way that no two of its edges intersect (except at a vertex on which both are incident)

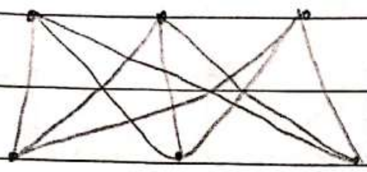
Scanned by CamScanner

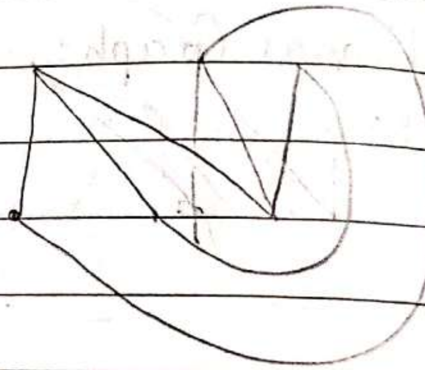
$K_{2,3}$





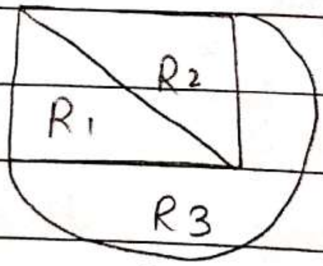
$K_{3,3}$





Planar representation with regions.

Ex:- K_4:



Planar representation of a graph splits the plane into regions

* Euler's Formula:

State: Let G be a connected planar simple graph with $|V| = v$, $|E| = e$. Let 'r' be the no. of regions in a planar representation of G then,

$$r = e - v + 2 \longrightarrow (1)$$

Proof: The proof is by induction on 'e'

Step1 → Suppose, if $e = 0$

$G: \quad \bullet v_1$

$v_1 = 1$ — Substitute in eq'n (1)

$e = 0$ — $1 = 0 - 1 + 2$

$r = 1$ — $1 = 1$. So it is true for $e = 0$

→ If $e = 1$

$G: \quad \underset{v_1}{\bullet} \!-\!-\!-\! \underset{v_2}{\bullet}$

$v = 2$; $e = 1$; $r = 1$

Substitute in equation (1)

$1 = 1 - \cancel{2} + \cancel{2} = 1$

∴ It is true for $e = 1$.

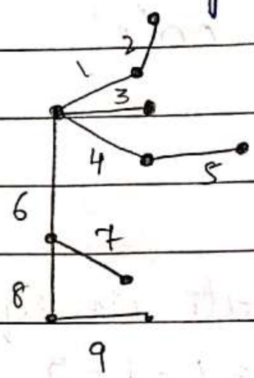
Step2: Hence, by mathematical induction let us assume that the result is true for every connected planar graph with 'k' edges where $0 \le e \le k$

i.e $r = k - v + 2 \rightarrow (2)$

Step 3:- We need to prove that the result is true for $e = k+1$

Case i: G_{K+1} has no cycles (so its a tree)

G_{K+1}:



$e = K+1$

$v = K+2$

$r = 1$

$1 = (\cancel{K}+1) - (\cancel{K}+2) + 2$

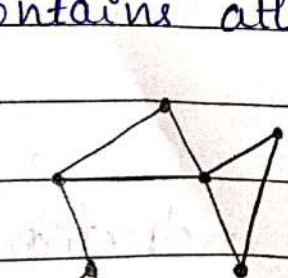
$1 = -1 + 2$

$1 = 1$

Hence, the result is true for $e = k+1$. So by mathematical induction the result is true for $\forall e$ (when it is a tree or acyclic graph)

Case 2: G_{K+1} contains atleast one cycle

G_{K+1}:



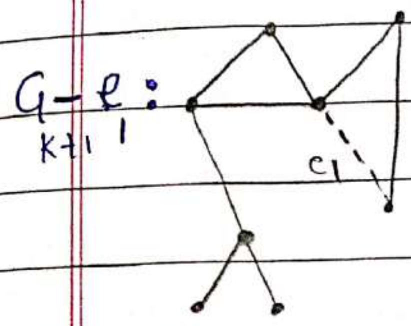
$e = K+1$

$v = v$

$r = r$

Consider an edge e_1 on the cycle & remove it from G_{K+1}

Deleting one edge from a cycle. The resulting graph $G_{k+1} - e_1$ contains,



$e = k+1-1$

$r = r-1$

$v = v.$

Consider,

$r = e - v + 2$

$r - 1 = k - v + 2$

$r = (k+1) - v + 2$

Hence, by the equation the result is true for $e = k+1$. So by mathematical induction it follows that the result is true for all non-negative integers 'e' (when it is a cyclic graph.)

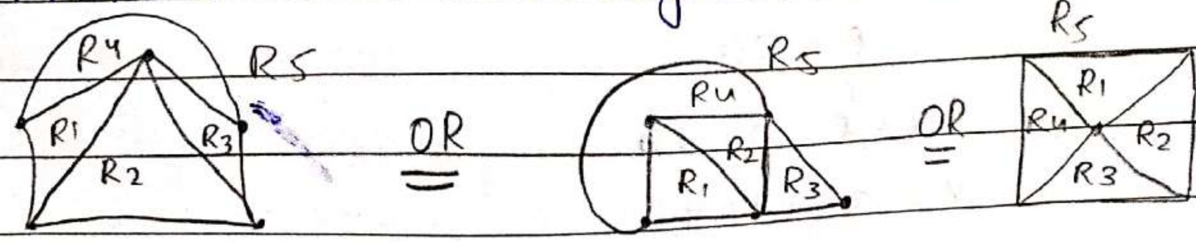
Examples:-

I. Show that the following graphs are palanar

(i) Graph of order 5 and size (12-4)

(ii) Graph of order 6 and size 12

(i) Order / vertices = 5   size / edges = 8



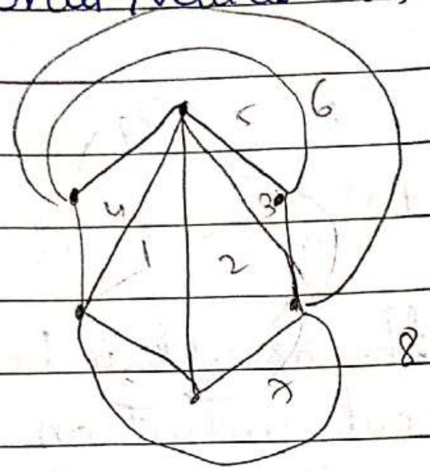
$r = 5;\ e = 8;\ v = 5$

$r = e - v + 2$

$5 = 8 - 5 + 2$

$5 = 5$

(iii) Order / vertices = 6; Edges = 12.



$r = e - v + 2$

$8 = 12 - 6 + 2$

$8 = 6 + 2$

$8 = 8$

2. A simple connected planar graph has 20 vertices each of degree 3. Into how many regions does a representation of this planar graph split the plane.

$v = 20;\ \deg(v) = 3$

$\sum \deg(v) = 2e$

$20 \times 3 = 2e$

$e = 30$

$r = e - v + 2$

$r = 30 - 20 + 2$

$r = 12$

Scanned by CamScanner

3. If G is simple connected planar graph with $v(\geq 3)$ vertices with $e \geq 2$ edges and r regions then

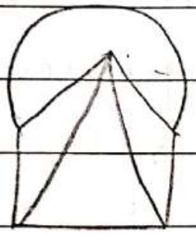
(i) e is always greater than equal to

$$e \geq \frac{3r}{2}$$

(ii) $e \leq 3v - 6$

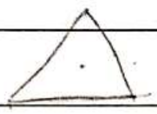
$4 \leq 3(4) - 6 \quad 4 \leq 6$

If G is triangle free graph then $e \leq 2v - 4$



$v = 5;\ e = 8 \quad (v) \geq 3$ and $e \geq 2$

$e \geq \frac{3r}{2}$ $\qquad$ $e \leq 3v - 6$



$8 \geq \frac{3(5)}{2}$ $\qquad$ $8 \leq 3(5) - 6$

$3 \geq \frac{3 \times 2}{2}$ $\qquad$ $8 \geq 7.5$ $\qquad$ $8 \leq 9$

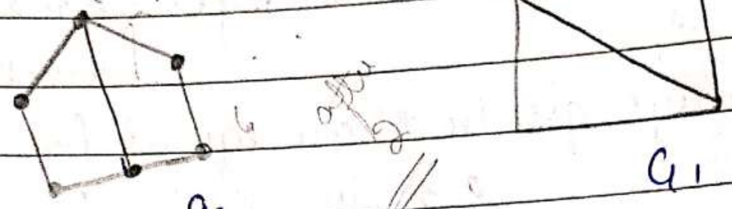
$3 \leq 3(3) - 6$ $\qquad$ // It is a triangle free graph

$3 \leq 3$

* Elementary Sub-division

If a graph is planar so will be any graph obtained by removing an edge $\{u, v\}$ and adding a new vertex w together with edges $\{u, w\}$ and $\{w, v\}$. Such an operation is called an elementary subdivision.

G:

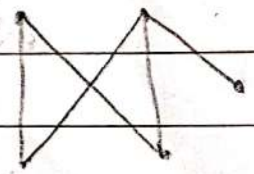


G_2 // G_1

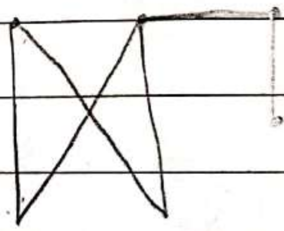
* Homeomorphic graphs:

The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called homeomorphic if they can be obtained from the same graph by sequence of elementary transformations

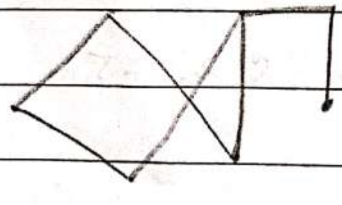
G_1:



G_2:



G_3:



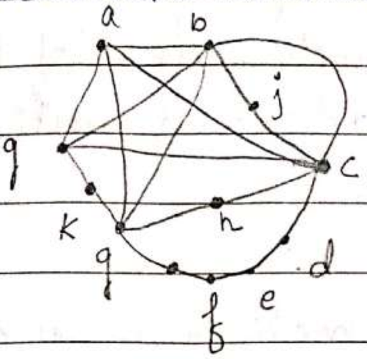
* Kuratowski's theorem

A graph is non-planar iff it contains a subgraph homeomorphic to $K(3,3)$ or K_5.

Scanned by CamScanner

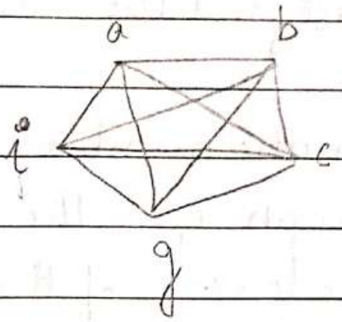
(P24* Determine whether the graph G is planar

Ex 1.

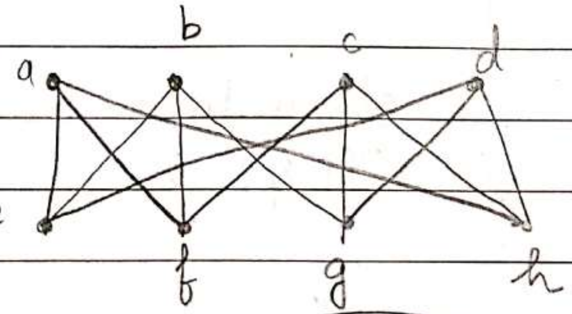


=

Delete vertex make adjacent

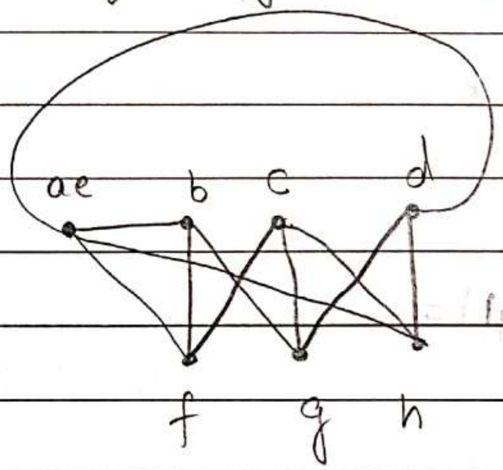


Ex2

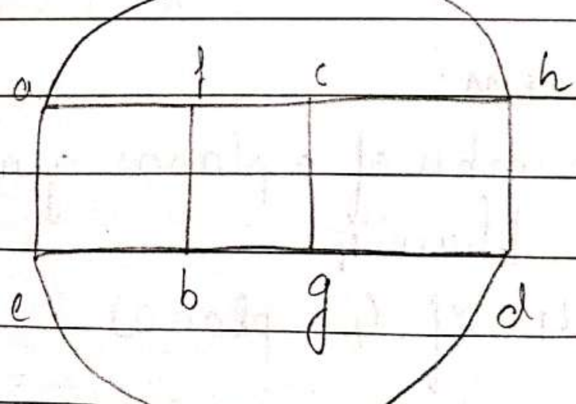


merging the vertices a&e

Sdn



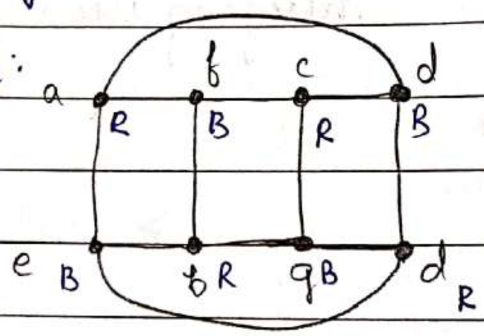
Ex3:



* **Coloring:-** A coloring of a simple graph is the assignment of color to each vertex of the graph so that no two adjacent vertices are assigned the same colour.

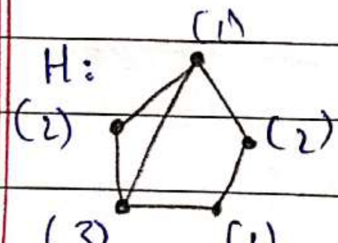
Chromatic number. The chromatic number of the graph G is the least no. of color needed for coloring of the graph and is denoted by $\chi(G)$

Ex 1: G:



Chromatic number.
$\chi(G) = 2$

Ex 2: H:

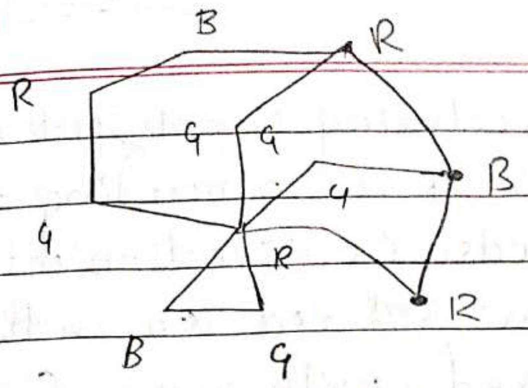


$\chi(H) = 3$

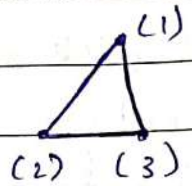
Four-color theorem:

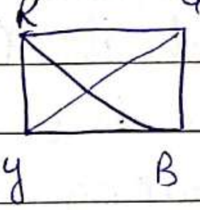
State: The chromatic number of a planar graph is not greater than 4

$$\chi(G) \leq 4 \text{ if } G \text{ is planar}$$



* Chromatic no. of K_n:

K_3: 

K_4: 

* Every vertex is adjacent to remaining vertices in the graph. *

$\chi(K_n) = n$, K_n is a complete graph with n vertices. Every pair of vertices are adjacent so no two vertices can be assigned a same color

Cn

Pn Chromatic number for cycle:- C_n even = 2; (n is even)
(verticles) C_n odd = 3; (n is odd)

Chromatic number for

What is the chromatic number of the graph C_n where $C \geq n$

→ Two colors are needed to color C_n when n is even

Scanned by CamScanner

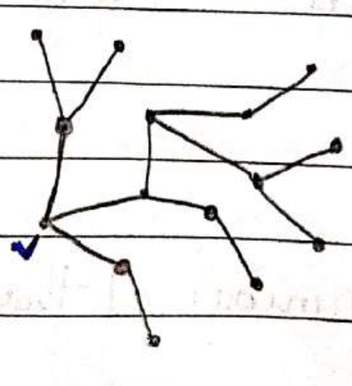
Bipartate: $\chi_n = 2$

to construct such a coloring simply pick a vertex and let us say colour it red proceeding around the graph in a clockwise direction coloring the 2nd vertex green 3rd red so on the nth vertex will be colored with green. So chromatic number is 2 for even vertices.

When n is odd the χ of C_n is 3 pick any vertex If you want to assign 2 colors alternative go on coloring in clock-wise direction. The nth vertex is adjacent to nth vertex and also n-1th vertex so both the colors can't be used to color the nth vertex. So nth vertex should be colored with third color. So χ of C_n is 3 if n is odd

Theorem: Prove that every tree with two or more colors is 2-chromatic

Proof:

T: 

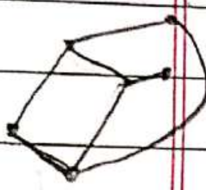
Select any vertex v in the given tree T. Consider T as a rooted tree at the vertex v then color v with color 1 then color all the vertices adjacent to v with color 2. The vertices adjacent to these vertices are colored using color 1. Continue this process till every vertex on T as being colored. Now on T we find that all vertices are even distance from v and including v have color 1 and all the vertices at odd distance from v have color 2

Now along any path on T the vertices are of alternating colors. Since there is one and only path b/w any 2 vertices on a tree. No 2 adjacent vertices have the same color. Then T is properly colored with two colors Hence tree is two chromatic.

NOTE: Every 2-chromatic need not to be a tree
/* Ex:- Cycle with even length / Bipartate Graph */

Theor: The graph with atleast one edge is two chromatin iff it has no circuits of odd length

Proof: Let 'G' be a connected graph with circuits of only even length. Consider a spanning tree on 'G'. Using coloring procedure and using earlier theorem. Let us properly color 'T' with two colors. Now add the (edges) chords to T one by one. Since 'G' had no circuits of odd length the end vertices of every chord being replaced are differently colored on 'T'. Does 'G' is colored with two colors with no adjacent vertices having the same color ∴ G is two chromatic



Conversely, if 'G' has a circuit of odd length we would need atleast 3 colors for that circuit which gives a contradiction. Hence 'G' has no circuits of odd length

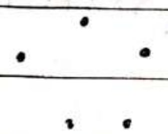
/* Since w.k.t. tree is two chromatic */

Chromatic Polynomial

Chromatic Polynomial Null graph:

Consider a null graph 'G' with 'n' vertices in this graph no two vertices are adjacent ∴ a proper coloring of this graph can be done by assigning a single color to all the vertices thus if there are λ colors each vertex of graph 'G' has λ

possible choices of colors assigned to it and as such the graph can be properly colored in λ^n different ways.

G:  $n=5$ $\therefore P(G:\lambda)=\lambda^5$

λ colours or λ^n in general

Complete graph:

$P(K_n:\lambda) = 0$, if $\lambda < n$

$= 1$, if $\lambda = n$.

$= \lambda(\lambda-1)(\lambda-2) -- (\lambda-(n-1))$ if $\lambda > n$

Consider a complete graph K_n if there are λ colors then no. of ways of properly coloring K_n is if $\lambda < n$ $p(K_n:\lambda) = 0$. Since the no. of colors are less than the no. of vertices. If $\lambda = n$ $P(K_n:\lambda) = 1$. there is exactly one way of coloring the complete graph if $n=\lambda$.

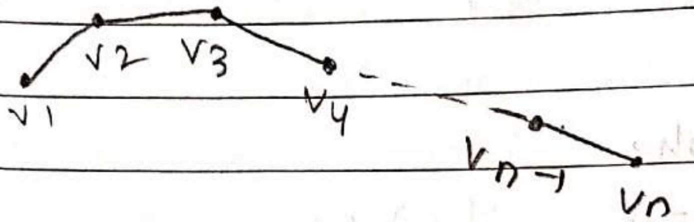
If $\lambda > n$ let $v_1, v_2 -- v_n$ be the vertices of K_n for a proper coloring of K_n a vertex v_1 can be assigned any of λ colors then vertex v_2 can be assigned any of the remaining $\lambda-1$ color then v_3 is assigned any of $\lambda-2$

colors and so on v_n is assigned $[\lambda-(n-1)]$ colors thus K_n can be properly colored in $\lambda(\lambda-1)\ldots[\lambda-(n-1)]$ ways.

$\therefore P(K_n:\lambda) = \lambda(\lambda-1)\ldots[\lambda-(n-1)]$ if $\lambda \geq n$

* Path (P_n)

P_n:

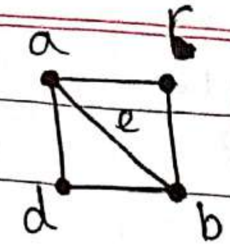


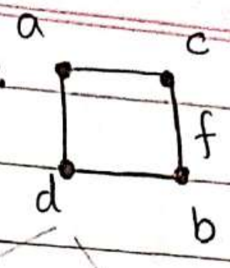
This graph is properly colored with two colors. Suppose $\lambda \geq 2$ for a proper coloring of the graph the vertex v_n can be assigned any one of λ colors and each of the remaining vertices can be assigned any one of $\lambda-1$ colors

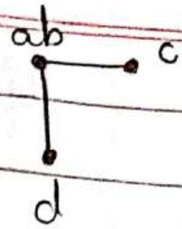
So $P(P_n:\lambda) = \lambda(\lambda-1)\ldots(\lambda-1)$

$= \lambda(\lambda-1)^{n-1}, \quad \lambda \geq 2$

NOTE:- $G=(V,E)$ be an undirected graph $e=(a,b) \in E$. Let $G_e = G-e$ be that subgraph of G which is obtained by deleting an edge e from graph 'G' without deleting the vertices a and b. Then a new graph G_e' is obtained by collising or (merging) the vertices a and b in the graph G_e

G:


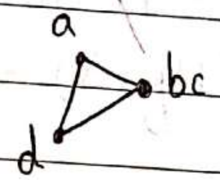
Ge:


Ge':


Decomposition theorem:-

If 'G' is a graph and 'e' is an edge of 'G' then

Hf:


Hf':


$$P(G:\lambda) = P(Ge:\lambda) - P(Ge':\lambda)$$

$$P(G:\lambda) = P(Ge:\lambda) - P(P_3:\lambda)$$

$$= \left[P(Hf:\lambda) - P(Hf':\lambda)\right] - P(P_3:\lambda)$$

$$= P(P_4:\lambda) - P(K_3:\lambda) - P(P_3:\lambda)$$

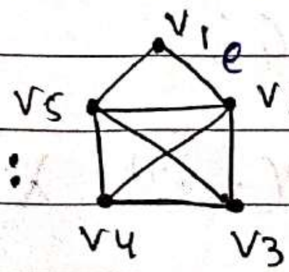
$$= \lambda(\lambda-1)^3 - \lambda(\lambda-1)(\lambda-2) - \lambda(\lambda-1)^2$$

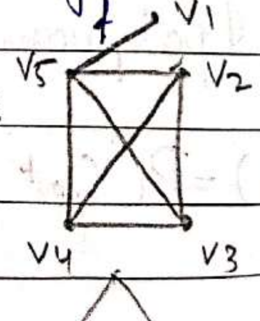
$$= \lambda(\lambda-1)\left((\lambda-1)^2 - (\lambda-2) - (\lambda-1)\right)$$

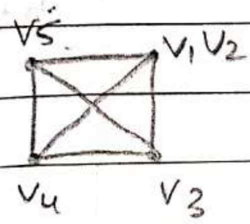
$$= \lambda(\lambda-1)(\lambda^2-4\lambda+4)$$

$$= \lambda(\lambda-1)(\lambda^2-2\lambda+1-\lambda+2-\lambda+1)$$

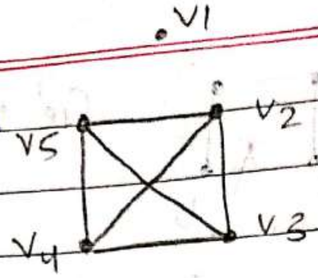
So it will be of degree 4 // 4 vertices.

Ex:- G:


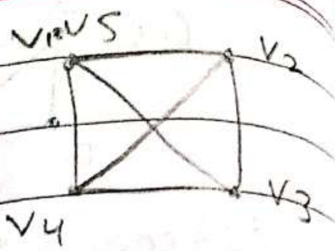
Ge:


Ge':


Hf:



Hf':



Ex:

By decomposition theorem

$P(G:\lambda) = P(G_e:\lambda) - P(G_e':\lambda)$

$= [P(H_f:\lambda) - P(H_f':\lambda)] - P(K_4:\lambda)$

$= P(H_f:\lambda) - P(K_4:\lambda) - P(K_4:\lambda)$

$= \underbrace{\lambda}_{v_1} \underbrace{[\lambda(\lambda-1)(\lambda-2)(\lambda-3)]}_{K_4} - 2(\lambda(\lambda-1)(\lambda-2)(\lambda-3)$

$= \lambda(\lambda-1)(\lambda-2)(\lambda-3)\,[\lambda-2]$

$= \lambda(\lambda-1)(\lambda-2)^2(\lambda-3)$

Alternate method for chromatin polynomial.

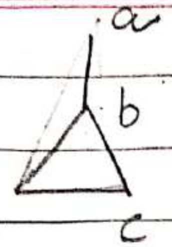
Let $G = (V, E)$ be a graph with $a, b \in V$ & $\{a, b\} = e \notin E$

Let G_e^+ denote the graph obtained by including e into G and G_e^{++} denoted the graph by merging the vertices a & b then the chromatin polynomial of G is

$$P(G:\lambda) = P(G_e^+:\lambda) + P(G_e^{++}:\lambda)$$

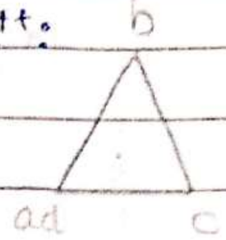
Scanned by CamScanner

Ex: G:

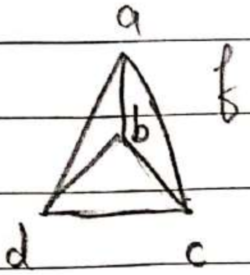


G_e^+:

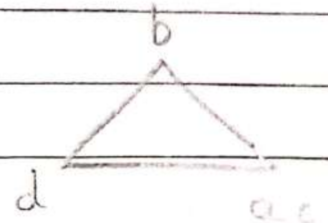
G_e^{++}:



H_f^+:



H_f^{++}:



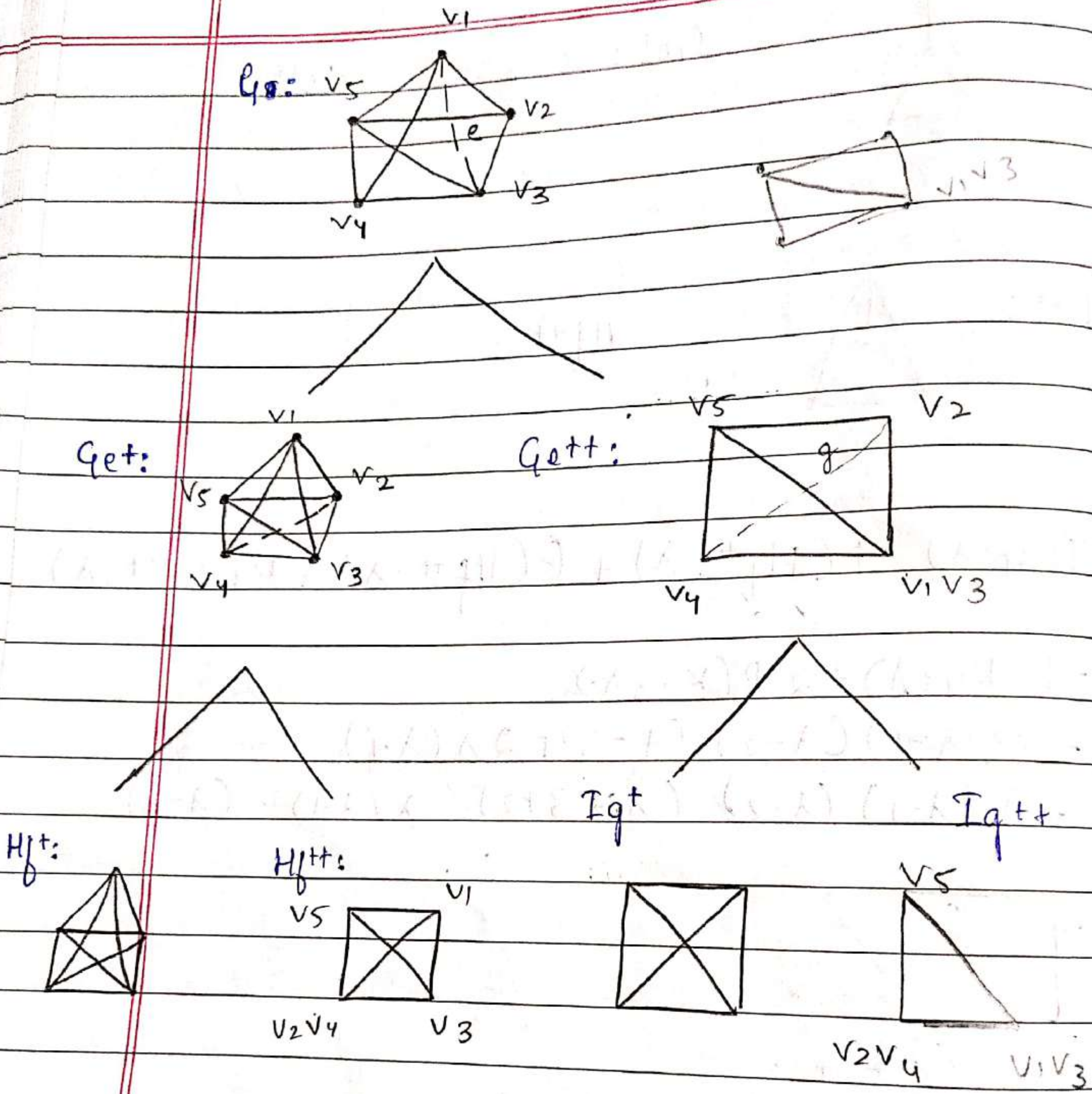
$$P(G:\lambda) = P(H_f^+:\lambda) + P(H_f^{++}:\lambda) + P(G_e^{++}:\lambda)$$

$$= P(K_4:\lambda) + 2P(K_3:\lambda)$$

$$= \lambda(\lambda-1)(\lambda-2)(\lambda-3) + 2\lambda(\lambda-1).(\lambda-2)$$

$$= \lambda(\lambda-1)(\lambda-2)(\lambda-3+2) = \lambda(\lambda-1)^2(\lambda-2)$$

Scanned by CamScanner



$$P(G:\lambda) = P(G_e^{+}:\lambda) + P(G_e^{++}:\lambda)$$

$$= [P(H_g^{+}:\lambda) + P(H_g^{++}:\lambda)] + [P(I_g^{+}:\lambda) + P(I_g^{++}:\lambda)]$$

$$= P(K_5:\lambda) + P(K_4:\lambda) + P(K_4:\lambda) + P(K_3:\lambda)$$

$$= \lambda(\lambda-1)(\lambda-2)(\lambda-3)(\lambda-4) + 2\lambda(\lambda-1)(\lambda-2)(\lambda-3$$

$\lambda(\lambda-1)(\lambda-2)$

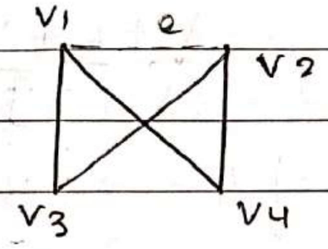
$= \lambda(\lambda-1)(\lambda-2)\left[(\lambda-3)(\lambda-4) + 2(\lambda-3) + 1\right]$

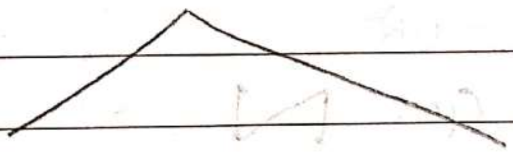
$= \lambda(\lambda-1)(\lambda-2)\left(\lambda^2 - 4\lambda - 3\lambda + 12 + 2\lambda - 6 + 1\right)$

$= \lambda(\lambda-1)(\lambda-2)\left(\lambda^2 - 7\lambda + 7 + 2\lambda\right)$

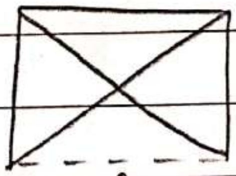
$= \lambda(\lambda-1)(\lambda-2)\left(\lambda^2 - 5\lambda + 7\right)$

(P 34) (iii) G:

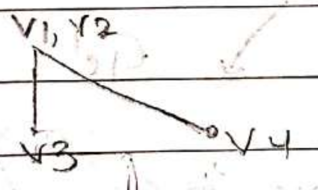




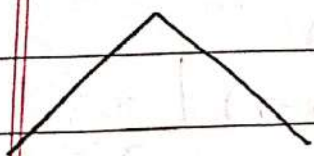
G_e^+:



G_e^{++}

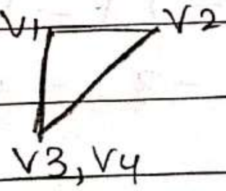


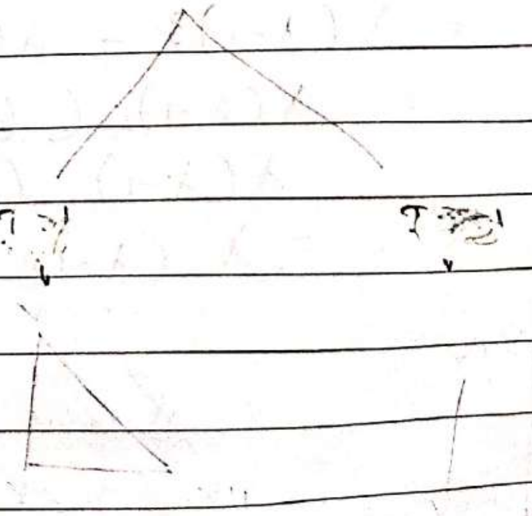
Path



H_f^+

H_f^{++}:





$$P(G:\lambda) = P(Ge^{+}:\lambda) + P(Ge^{++}:\lambda) + [P(P_3:\lambda)]$$

$$= [P(H_1^{+}:\lambda) + P(H_1^{++}:\lambda)] + P(P_3:\lambda)$$

$$= [P(K_4:\lambda) + P(K_3:\lambda)] + \lambda(\lambda-1)(\lambda-2) + \lambda(\lambda-1)^2$$

$$= \lambda(\lambda-1)(\lambda-2)(\lambda-3) + \lambda(\lambda-1)(\lambda-2) + \lambda(\lambda-1)^2$$

$$= \lambda(\lambda-1)[(\lambda-2)(\lambda-3) + (\lambda-2) + \lambda-1)]$$

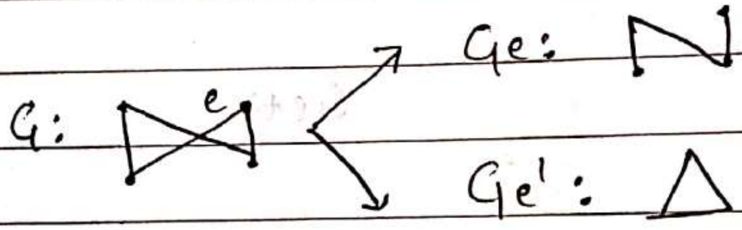
$$= \lambda(\lambda-1)[\lambda^2 - 3\lambda - 2\lambda + 6 + \lambda - 2 + \lambda - 1)]$$

$$= \lambda(\lambda-1)[\lambda^2 - 5\lambda + 6 + \lambda - 2 + \lambda - 1)$$

$$= \lambda(\lambda-1)[\lambda^2 - 3\lambda + 3].$$

// Order of G = n = 4 vertices.

– OR –



(Pu)

$$P(G:\lambda) = P(Ge:\lambda) \; P(Ge':\lambda) \; (K_3)$$

$$= \lambda(\lambda-1)^3 - \lambda(\lambda-1)(\lambda-2)$$

$$= \lambda(\lambda-1)((\lambda-1)^2 - (\lambda-2)]$$

$$= \lambda(\lambda-1)(\lambda^2 + 1 - 2\lambda - \lambda + 2)$$

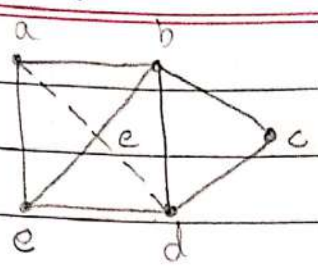
$$= \lambda(\lambda-1)(\lambda^2 - 3\lambda + 3)$$

Same polynomial

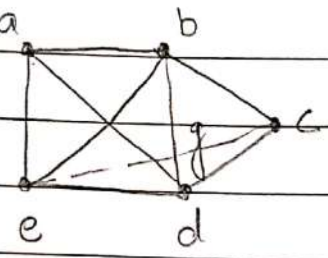
LP 27). K_5, $K_{3\text{-}3}$ // non-planar

G:

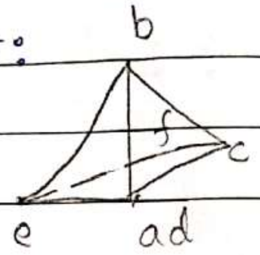
3.4(ii)



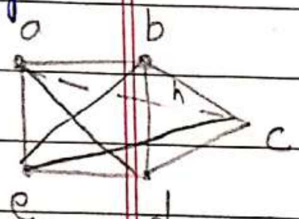
Ge+



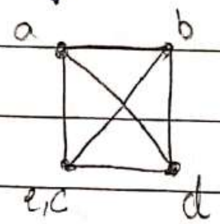
Ge++:



Ag+

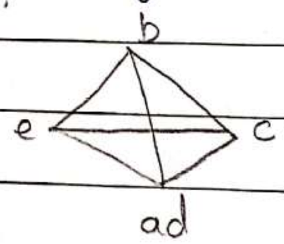


Ag++:



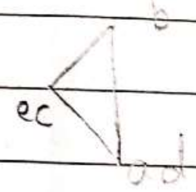
K4

Hf+



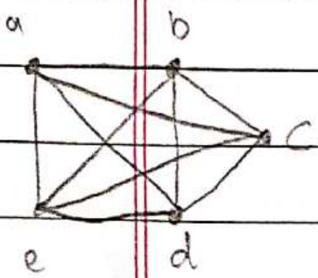
K4

Hf++



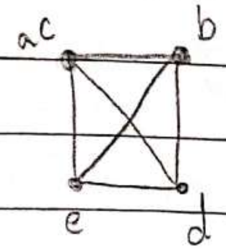
K3

Ih+



K5

Ih++



K4

$P(G:\lambda) = P(Ge^{+}:\lambda) + P(Ge^{++}:\lambda)$

$= [P(Ag^{+}:\lambda) + P(Ag^{++}:\lambda)] + [P(Hf^{+}:\lambda) + P(Hf^{++}:\lambda)]$

$= [P(In^{+}:\lambda) + P(In^{++}:\lambda)] + P(K_4:\lambda) + P(K_4:\lambda) + P(K_3:\lambda)$

$= P(K_5:\lambda) + P(K_4:\lambda) + P(K_4:\lambda) + P(K_4:\lambda) + P(K_3:\lambda)$

$= P(K_5:\lambda) + 3P(K_4:\lambda) + P(K_3:\lambda)$

$= \lambda(\lambda-1)(\lambda-2)(\lambda-3)(\lambda-4) + 3\cdot\lambda(\lambda-1)(\lambda-2)(\lambda-3) + \lambda(\lambda-1)(\lambda-2)$

$= \lambda(\lambda-1)(\lambda-2)$

Theorem: A graph with n vertices is a complete graph iff its chromatic polynomial is
$P_n(\lambda) = \lambda(\lambda-1)(\lambda-2) \text{ ------ } [\lambda-(n-1)]$

With λ colors there are λ different ways of coloring any selected vertex of a graph. A second vertex can be colored properly using exactly $\lambda-1$ ways. The third in $\lambda-2$ ways, the fourth in $\lambda-3$ ways and so on the nth vertex in $[\lambda-(n-1)]$ ways iff every vertex is adjacent to every other vertex iff the graph is complete.

Theorem: A n vertex graph is a tree iff its chromatic polynomial is
$$P_n(\lambda) = \lambda(\lambda-1)^{n-1}$$

Proof:- The proof goes with induction on 'n'

Consider, $n=1$

T:

$P_1(\lambda) = \lambda$

Consider $n=2$

T: 

$P_2 = \lambda(\lambda-1)$

So it is true for 1, 2, 3.

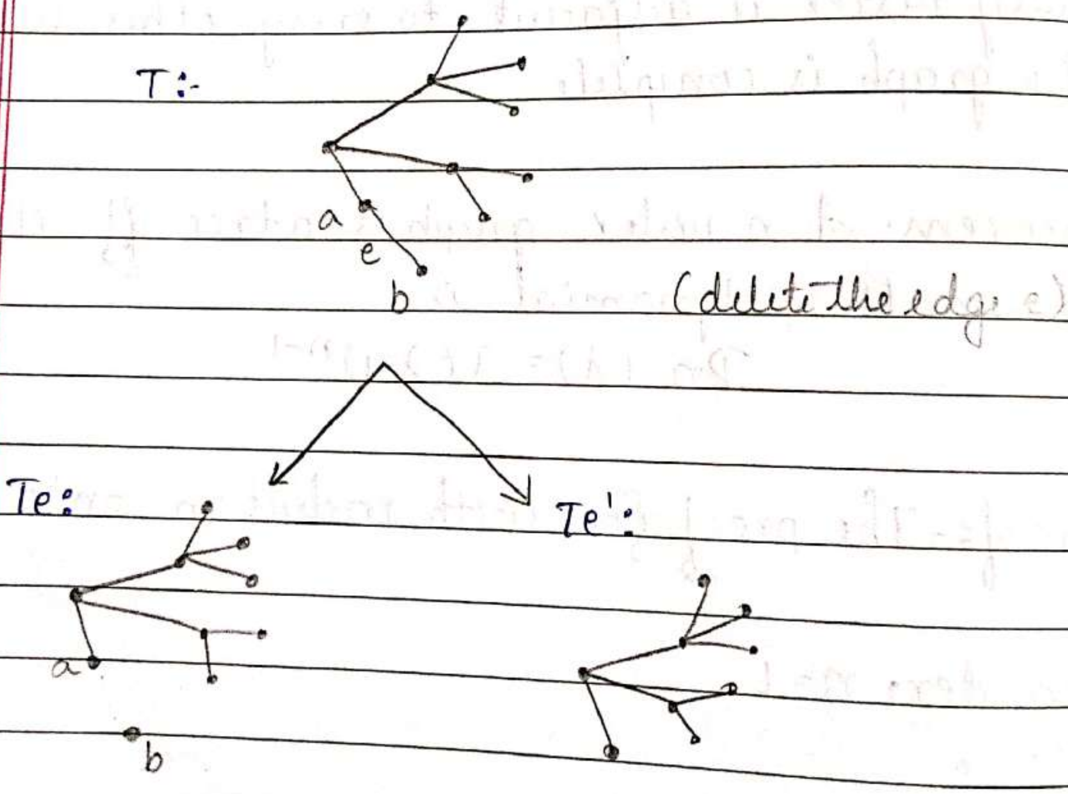
all tree with

Let us assume that the result is true for 'k' or fewer vertices ($n \leq k$)

Let 'T' be the tree with k+1 vertices. Since 'T' has two pendent vertices. Let 'e' be an edge to one of these leaves $\therefore$ By decomposition theorem,

$$P(T:\lambda) = P(Te:\lambda) - P(Te':\lambda)$$

where Te is an isolated vertex together with a tree on k vertices and Te' is a tree on k vertices



$$P(Te:\lambda) = \lambda\,(\lambda\,(\lambda-1)^{k-1})$$
$$P(Te':\lambda) = \lambda\,(\lambda-1)^{k-1}$$

$$P(T:\lambda) = \lambda\,(\lambda(\lambda-1)^{K-1}) - \lambda(\lambda-1)^{K-1}$$

$$= \lambda(\lambda-1)^{K-1}[\lambda-1]$$

$$= \lambda(\lambda-1)^{K-1}(\lambda-1)$$

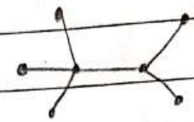
$$= \lambda(\lambda-1)^{K-1+1}$$

$$= \lambda(\lambda-1)^{K}$$

Hence the result is true for $k+1$ vertices $\therefore$ by mathematical induction the result is true for all 'n'

TREES

def'n: A tree is connected graph without any circuits cycles.

T: 

Theorem: A graph is a tree iff there is exactly one path between every pair of its vertices.

Let 'G' be the graph and let there exist exactly one path between every pair of vertices in 'G'. Therefore 'G' is connected. Now 'G' has no cycles because if 'G' contains a cycle say b/w the vertices 'u' and 'v' then there are two distinct paths between u and v

Hence 'G' is connected and is without cycle
∴ 'G' is a tree

Conversely, let 'G' be a tree since 'G' is connected there is atleast one path b/w every pair of vertices in G. Let there be two distinct paths between two vertices 'u' and 'v' in G. Then the union of these two paths contain a cycle which contradicts to the fact that 'G' is a tree. Hence there is exactly one path between every pair of vertices of a tree

Theorem:
A tree with 'n' vertices has (n-1) edges

Proving it by induction on 'n'

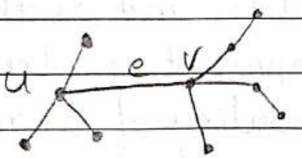
Consider, $n=1$ T/G: $e=0$

" $n=2$ T/G: $e=1$

" $n=3$ T/G: $e=2$

The result is true for some smaller vertices

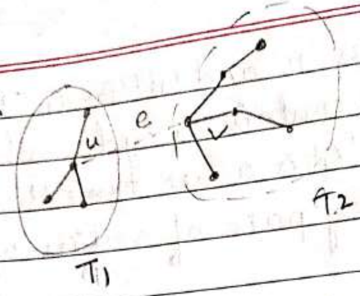
Assuming the result is true for all trees with fewer vertices than n vertices $(k<n)$.



Remove 'e'

Let 'T' be a tree be with 'n' vertices and 'e' be an edge with end vertices 'u' and 'v' So the only theorem:-

T - e :



No. of edges in T = no. of edges in T_1 + no. of edges T_2 + e

$= n_1 - 1 + n_2 - 1 + 1$

$= n_1 + n_2 - 1$

$= n - 1$

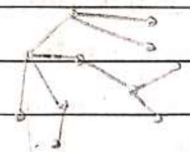
Hence true for all n. Hence by mathematical induction it is true for all trees with n vertices

path between 'uv' is 'e' ∴ deletion of e from disconnects 'T'. Now T-e consists of two components T_1 and T_2 and as there are no cycles in 'T' each component is a tree. Let n_1 and n_2 be the no. of vertices of T_1 and T_2 respectively such that $n_1 < n$ and $n_2 < n$ and $n_1 + n_2 = n$
∴ By induction hypothesis, the no. of edges in T_1 is $n_1 - 1$ and no. of edges in T_2 is $n_2 - 1$.

Prove that every non-trivial tree has atleast two vertices of degree 1 -OR-
Prove that in a tree with two or more vertices there are atleast two leaves.

Consider a tree 'T' with 'n' vertices $n \geq 2$ then it has $n-1$ edges $\therefore$ The the no. of edges with 'n' vertices is $n-1$. Then by handshaking theorem we have,

T:



$$\sum_{i=1}^{n} d_i = 2e$$

$$d_1 + d_2 + \dots d_n = 2e = 2(n-1) = 2n-2.$$

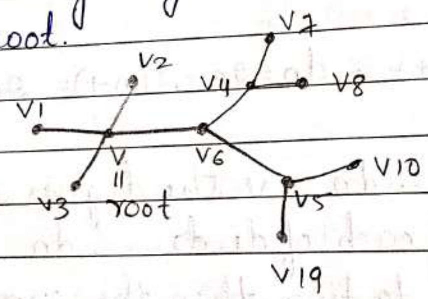
where $d_1, d_2 \dots d_n$ are the degrees of $v_1, v_2 \dots v_n$ respectively. If each of $d_1, d_2 \dots d_n$ are greater than or equal to two then the sum must be greater than or equal to $2n$. Since this is not true because the summation of d_i's is $2n-2$ $\therefore$ there must be d_i's whose degree is less than two since 'T' is connected no d_i's can be zero. Without loss of generality let us take the degree $d_1 = 1$

$d_2 + d_3 + \dots + d_n = 2n-2-d_1$
$= 2n-2-1$
$= 2n-2-1 = 2n-3$

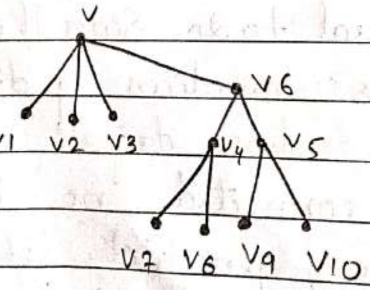
$\therefore \Sigma d_i = 2n-2$ is possible only if there is one more d_i's is equal to 1 $\therefore$ In 'T' there are atleast two vertices with degree one or leaves.

Rooted tree: A rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root.

T:



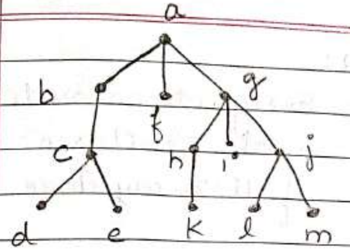
T:



LP 7: Parent, child, sibling, ancestors, descendants, leaf, internal node.

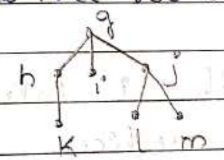
internal vertices = root

classmate
Date
Page



(i) parent of c = b
(ii) children g = h, i, j
(iii) sibling of h = i, j
(iv) descendants of b = c, d, e.
(v) ancestors of c = b, a
(vi) " " k = h, g, a
(vii) internal vertices = a, b, c, g, h, j
(vii) leaves :- d, e, k, l, m, i, f

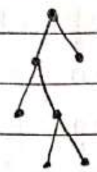
(viii) sub-tree rooted at g



* M-RAY TREE
A rooted tree is called M-RAY Tree if every internal vertex has no more than 'm' children

* FULL M-RAY TREE
The tree is called full M-RAY tree if every internal vertex has exactly m children (or no children).
A m-ray with m=2 is called binary Tree

LP9:- T1:



T2:

Every internal vertex is with 3 children

full 3-ray tree

full 2-ray or binary

T3:

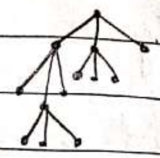
Full 5-ray tree

T4: Some of internal vertices has 2 children and some has 3 children. So it is not full m-ray tree

Theorem: A full m-ray tree with i internal vertices contains $n = mi + 1$ vertices

LP10

Every vertex except the root is the the child of internal vertex i.e. each of the i internal vertice has ~~[illegible]~~ exactly 'm' children $\therefore$ mi vertices in the tree other than the root $\therefore$ Including the root the the tree contains $n = mi + 1$ vertices

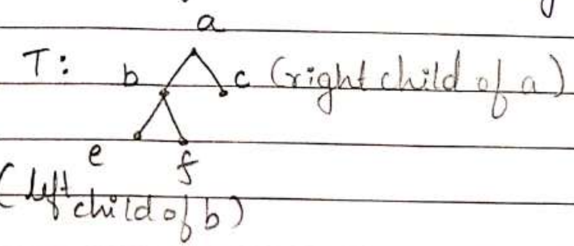


$mi + 1 = n$

$4(3) + 1 = 13$

Binary Tree:-

A binary tree is a rooted tree in which each vertex has atmost two children.
Each child in a binary tree is designated either a left child or a right child.



Full binary tree is a tree in which every internal vertex has exactly two children.

// pendent vertices $i+1$ // total is $2i+1$
and i internal then $2i+1-i = i+1$
pendent vertices

A full binary tree has 20 leaves. How many vertices does it have.

No. of leaves $= 20 = i + 1$

$i = 19$

No. of vertices $= 2 \times i + 19 = 2 \times 19 + 1 = 39$ vertices

Find the number of leaves in a full binary tree if it has 29 vertices

$29 = 2i + 1$

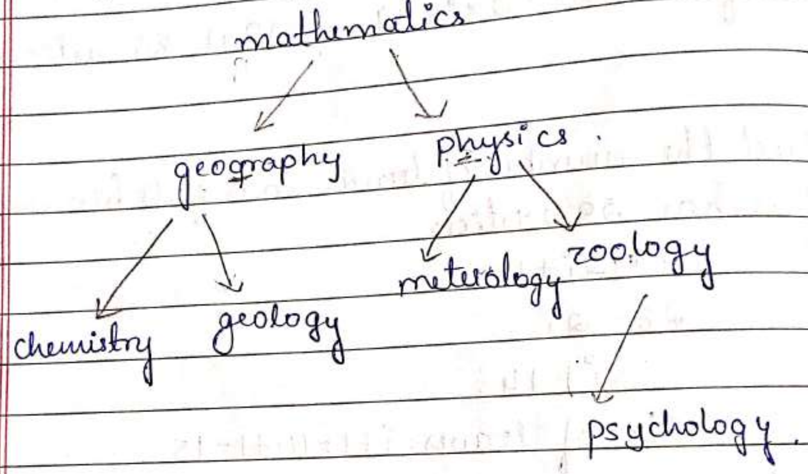
$28 = 2i$

$i = 14.$

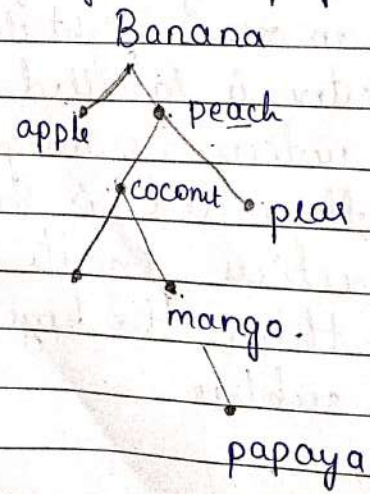
No. of leaves $= i + 1 = 14 + 1 = 15$

Binary Search tree. Binary search tree which is a binary tree in which each child of a vertex is designated as a right or left child. No vertex has more than one right child or left child and each vertex is labelled with a key. Furthermore vertices are assigned keys so that the key of the vertex is both larger than keys of all vertices in its left sub-tree and smaller than the keys of all vertices in its right subtree

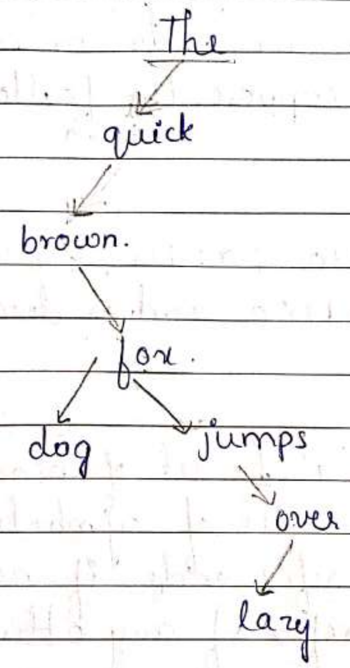
(P 15) Form BST for the words mathematics, physics, metiorology, geology, geography, zoology, psychology, chemistry



i.P 16) Banana, peach, apple, ~~[illegible]~~, pear, coconut, mango and papaya.



"The quick brown fox jumps over the lazy dog"



Prefix codes:- A sequence consisting of only 0 and 1 is called binary sequence. Binary sequences are used as codes for msgs sent through transmitting channels. Suppose we use the following coding scheme for these letters

a:1 e:0 n:10 r:01 t:101

Under this coding system suppose the msg eat is to be transmitted then the

classmate
Date
Page

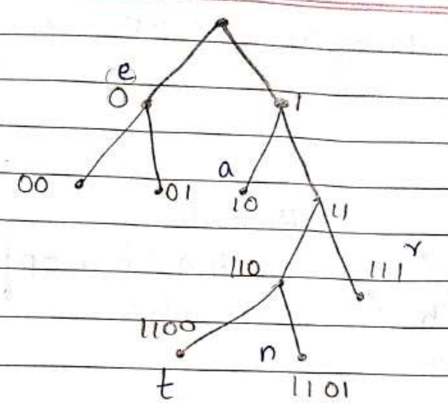
coded form of the msg is given by the binary sequence 01101. The sequence 01101 can be decoded not only as eat but also as 'rt' or 'eana' or 'eaar' or 'rar' or 'rna'.
So alternatively suppose the following coding scheme.

a:10, e:0, n:1101, r:111, t:1100.
then 'eat' = 0101100 and is decoded as only "eat"

Prefix code: Let 'T' be set of binary sequences that represent a set of symbols then 'T' is called as prefix code if no sequence in p is the prefix of any other sequence in p

$P_1 = \{10, 0, 1101, 111, 1100\}$
$P_2 = \{\underset{0}{\underline{000}}, \underset{1}{\underline{001}}, \underset{1}{\underline{01}}, \underset{2}{\underline{10}}, \underset{3}{\underline{11}}\}$
prefix code.

This can be represented as binary tree



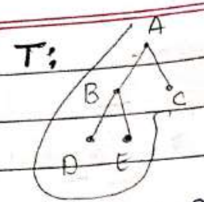
* Tree traversal:-

Ordered rooted trees are often used to store information. We need procedures for visiting each vertex of an ordered rooted tree to assess the data. We will describe several important algorithms for visiting all the vertices of an ordered rooted tree.

Preorder traversal. Let 'T' be an ordered rooted tree with root 'r'. If 'T' consists only 'r' then 'r' is the preorder traversal of 'T'. Otherwise suppose $T_1, T_2, \ldots, T_n$ are the sub-trees at 'r' from left to right. The preorder traversal begins by visiting 'r'. It continues by traversing T_1 in preorder, then T_2 in preorder and so on until T_n is traversed by preorder.

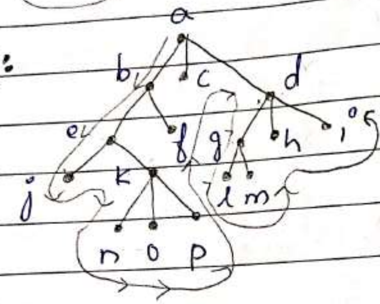
Date
Page

T:



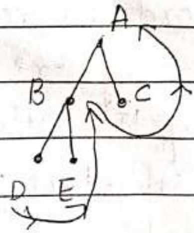
Preorder : ABDEC

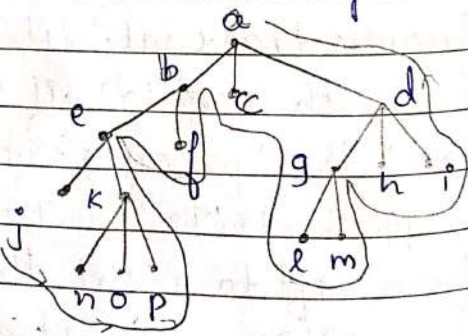
Q:



Preorder:
a b e j k n o p f c d g l m h i

Postorder traversal: Let 'T' be the ordered tree with root 'r' if 'T' consists of only 'r' then r is the postorder traversal of 'T' otherwise Suppose that $T_1, T_2 \ldots T_n$ are the subtree at 'r' from left to right the postorder traversal begins from sub-tree T_1 in postorder then T_2 in postorder so on T_n in postorder and ends by visiting 'r'

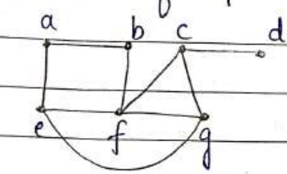




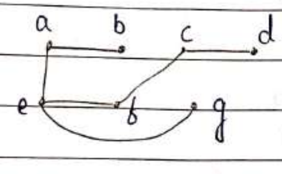
postorder: DEBCA

Spanning Trees:- Let 'G' be a connected graph a subgraph 'T' of 'G' is called as spanning tree of 'G' if 'T' is a tree and contains all the vertices of 'G'

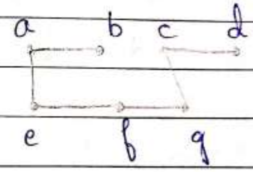
Ex:- G:



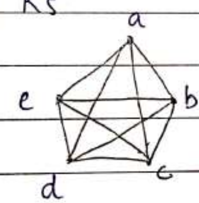
T:



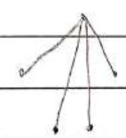
-OR-



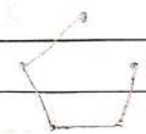
LP 31: 1) K5



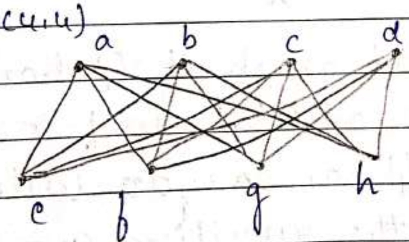
T:



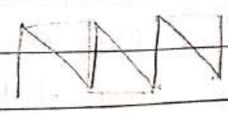
-OR-



2) K(4,4)

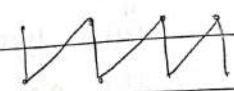


T:



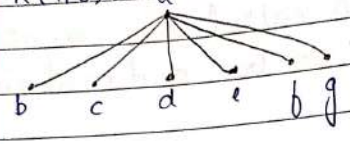
-OR-

T:



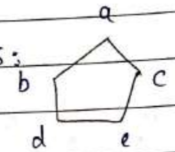
/* even cycle */

K(1,6):



It itself is a spanning tree

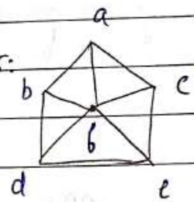
C5:

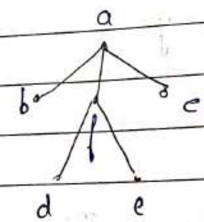


T1:

T2:

W5:





Theorem:

State:- A graph is connected iff it has a spanning tree

Proof:- Let 'G' be a connected graph if 'G' has no cycles then 'G' is a tree and 'G' itself is a spanning tree of 'G'. If 'G' has cycles delete an edge from each cycle the resulting graph G' is cycle free connected and contains all vertices this graph G' is a spanning tree of 'G'. Thus 'G' has a spanning tree

Conversely, suppose a graph 'G' has a spanning tree 'T' since 'T' is a tree there exists a path between every pair of vertices in 'T'. Since 'T' is a spanning tree 'T' contains all vertices of 'G' so there is a path b/w every pair of vertices in 'G' Hence 'G' is connected.

Theorem. Show that Hamilton path is a spanning tree

Proof. A Hamilton path 'p' in a connected graph 'G' if such path exists is a path that contains every vertex of 'G' that if 'G' has n vertices then the path 'p' has n-1 edges. Then 'p' is a connected sub-graph of 'G' with n vertices and n-1 edges ∴ 'P' is a tree since it contains all the vertices of 'G' be is a spanning tree of 'G'

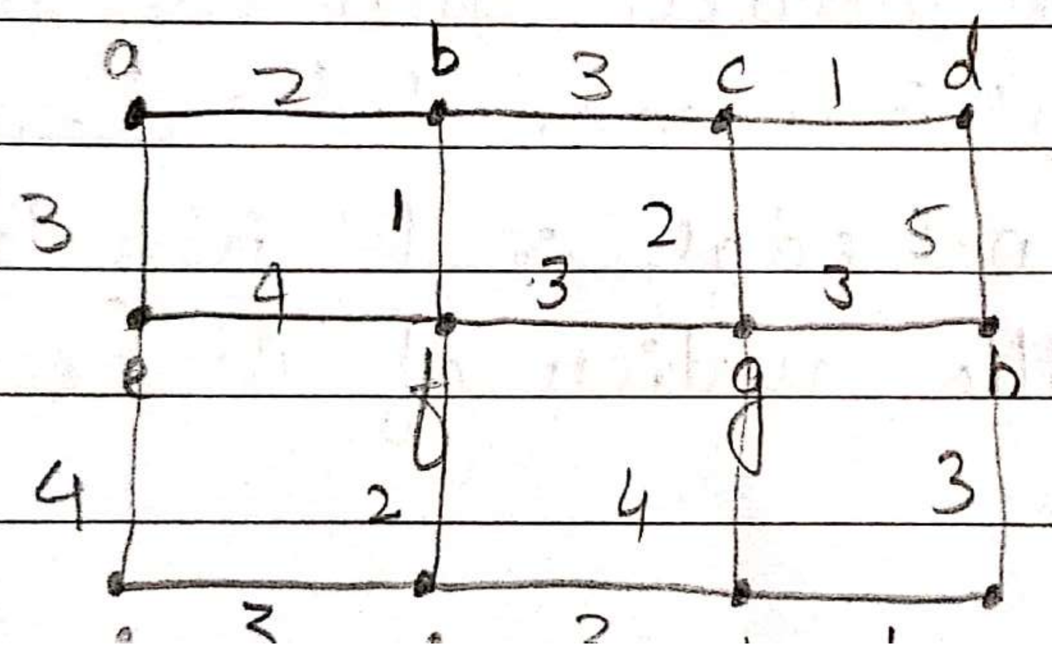
Weighted Graph: Let 'G' be a graph and suppose there is a +ve real number associated with each edge of 'G'. then 'G' is called weighted graph

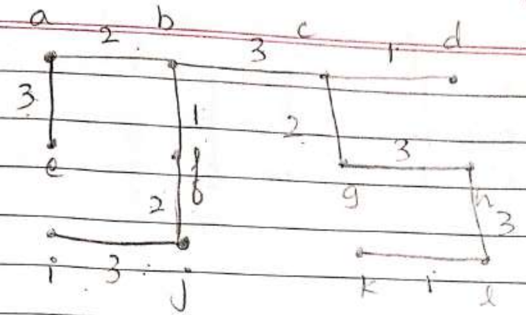
Minimal Spanning tree

A spanning tree whose weight is the least is called the minimal spanning tree of the graph

Starting with a smallest weighted edge proceed sequentially by selecting one edge at a time such that no cycle is formed

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





Weight of spanning tree = 24.

//all are connected so stop

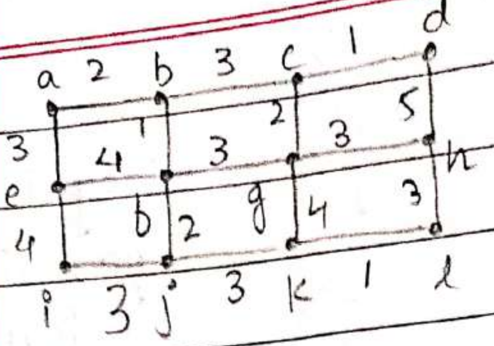
* Prim's Algorithm:

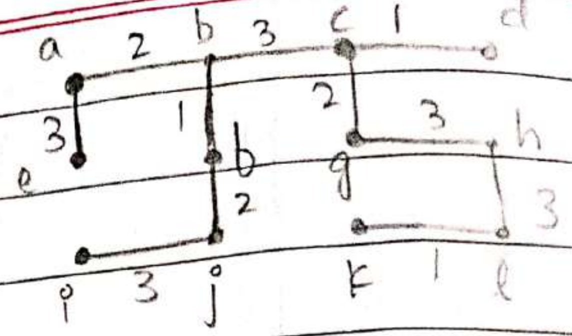
To carry out Prim's Algorithm begin by choosing any edge with smallest weight putting it into a spanning tree. Successively add to the tree the edges of min weight that are incident to a vertex already in tree and not forming a cycle with those edges already in the tree. Stop when (n-1) edges have been added.

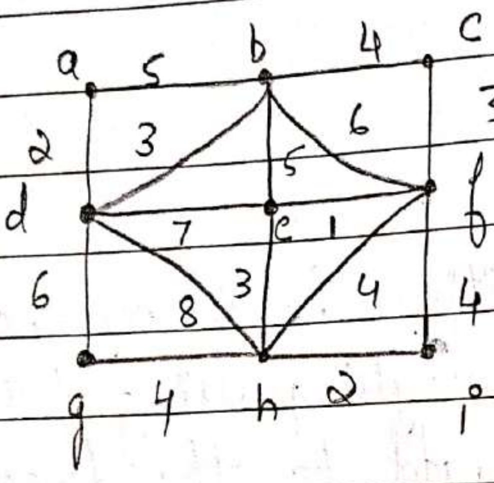
Choice	1	2	3	4	5	6	7	8	9	10	11
Edges	bf	ab	fj	ij	bc	ae	cd	cg	gh	hl	kl
Weight	1	2	2	3	3	3	1	2	3	3	1

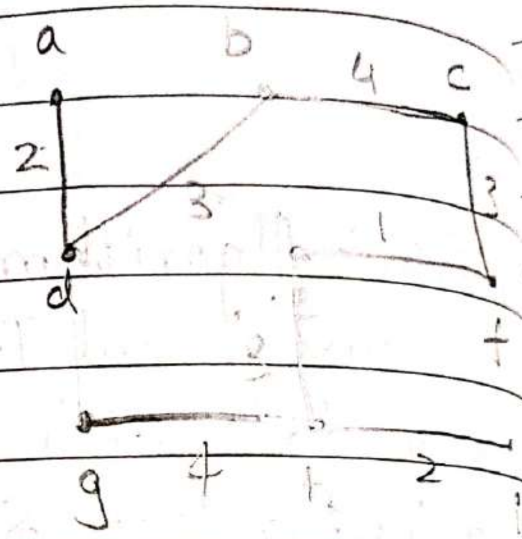
Total Weight = 24.

Scanned by CamScanner







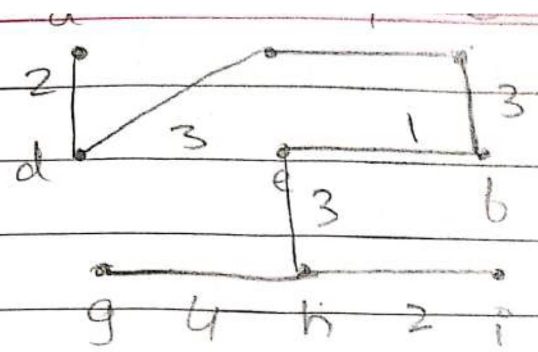


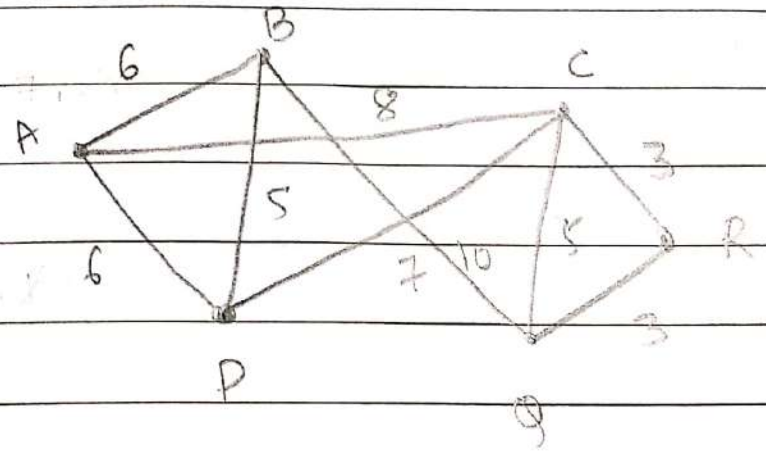
Ex:-

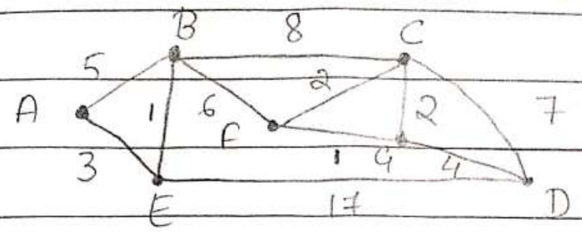
Weight = 22.

Edge : ef ad hi db cf eh bc fi g
ab be dg bf de dh

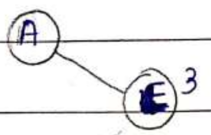
Weight: 1 2 2 3 3 3 4 4 4


2
3
1
3
d
e
3
b
g
4
h
2
i


B
6
C
8
A
3
5
R
6
7
10
5
3
P
Q

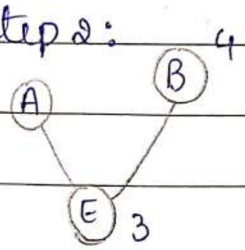


Step1:-



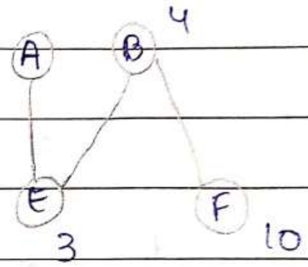
S = {A, E}

step2:



S = {A, E, B}

step3:



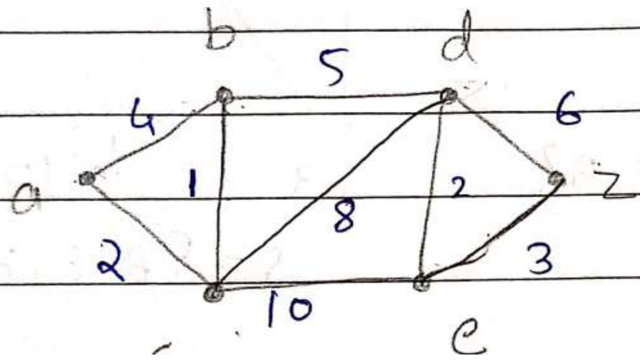
S = {A, E, B, F}

Step4: 4

Step5:

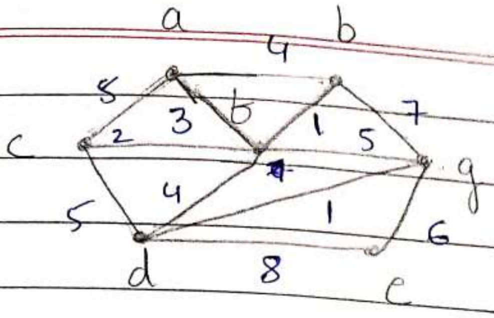
(D) 15

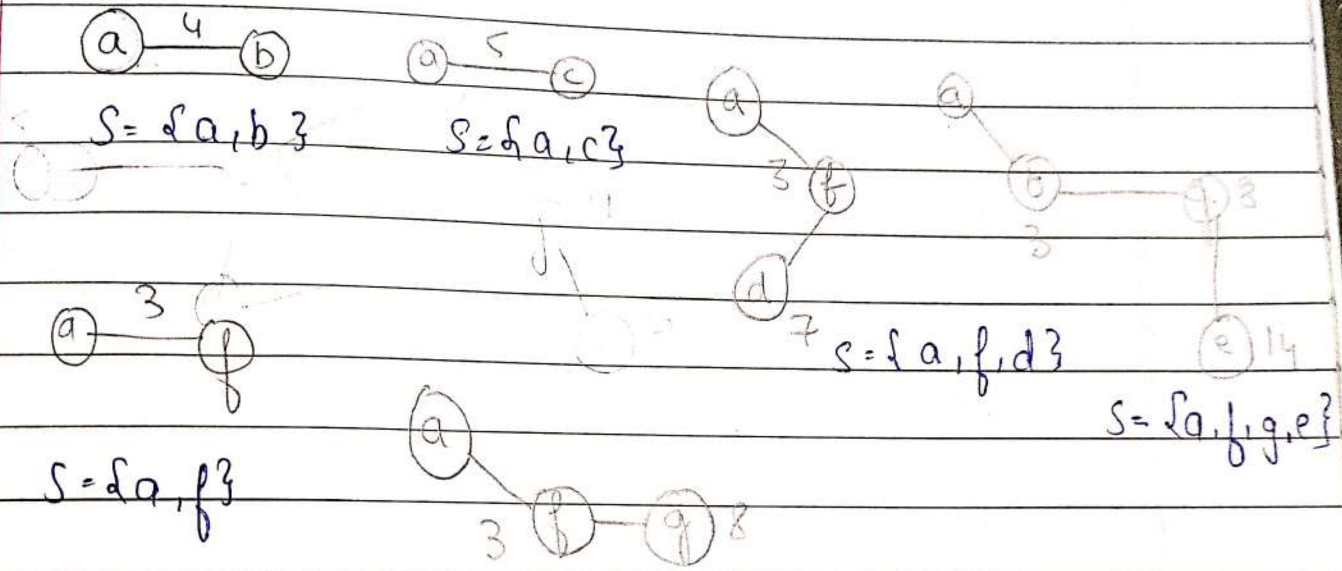
S = {A, E, B, F, G, D}



Scanned by CamScanner

IP:-





S = {a, b}

S = {a, c}

S = {a, f, d}

S = {a, f, g, e}

S = {a, f}

S = {a, f, g}.