i.e a with d and a with b is adjacent vertex and b and d both ~~a~~ vertex with degree 3

$\Rightarrow$ G and H are not isomorphic (becau we can't find f(a)).

Euler and Hamilton Paths :-

Euler Circuit and Euler path:-

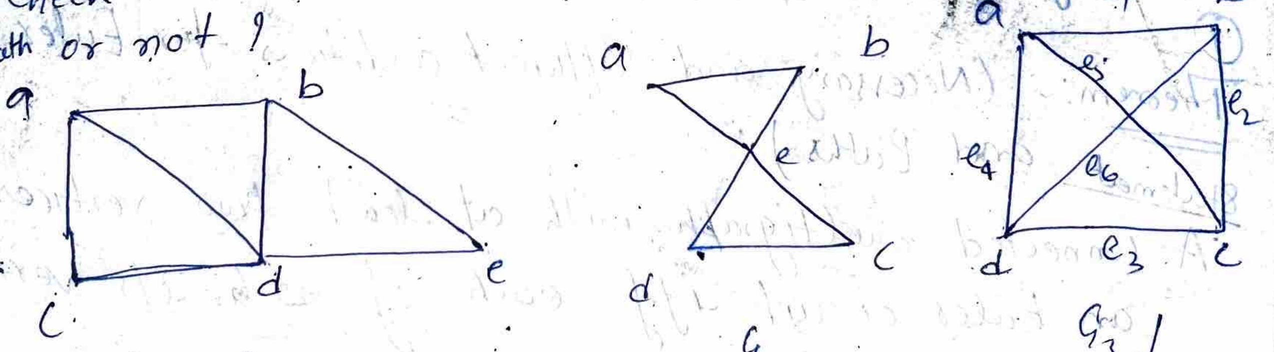
Defⁿ An euler circuit in a graph G is a simple circuit (i.e edges not repeated) containing every edge of G.

An euler path in G is a simple path containing every edge of G.

Use → the problem of traveling across every bridge without crossing any bridge more than once can be rephrased in terms of this model.

i.e Is there a simple circuit in ~~a~~ any multigraph that contains every edge?

Ex:- Check whether the following UGs have an euler circuit and euler path or not?

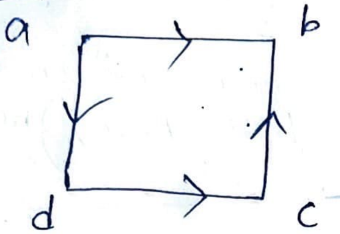


G_1: a – b – e – d – c – a – d
↓ euler path
but it does not have euler circuit

G_2: a – b – e – c – d – e – a
↓
euler circuit
but does not have euler path

G_3: $a \xrightarrow{e_1} b \xrightarrow{e_2} c \xrightarrow{e_5} a \xrightarrow{e_4} d \xrightarrow{e_6} b \xrightarrow{e_2} c \xrightarrow{e_3} d$
↓
~~does not have eule~~
neither have euler path nor euler circuit

Same for DG

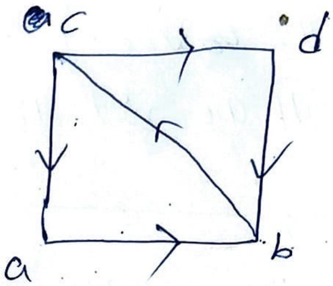


a – d – c – b

or a – b ↓ H_1

neither have euler circuit nor path

(2)

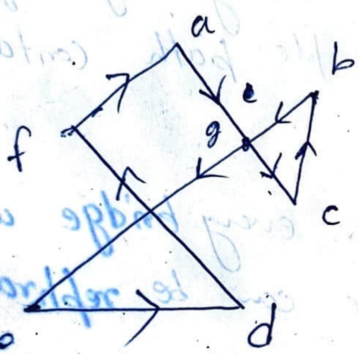


H_3

c – a – b – c – d – b

↓ euler path but not euler circuit.

(3)



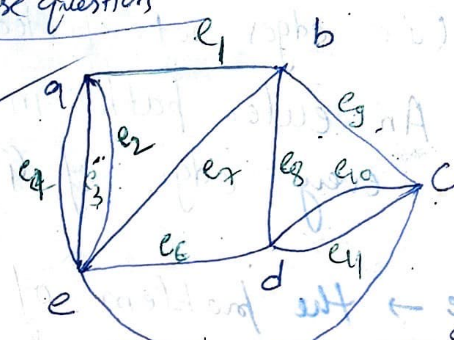
H_2

euler circuit

e – d – f – a – g – c – b – g – e

or a – g – c – b – g – e – d – f – a

Excercise question



every vertex has even degree so by theor graph has an euler circuit

euler circuit

$b \xrightarrow{e_1} a \xrightarrow{e_2} e \xrightarrow{e_3} a \xrightarrow{e_4} e \xrightarrow{e_5} c \xrightarrow{e_6} d \xrightarrow{e_6}$
$\xrightarrow{e_7} b \xrightarrow{e_8} d \xrightarrow{e_{10}} c \xrightarrow{e_9} b \to EC$

(1)

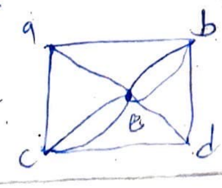
Theorem:- (Necessary and Sufficient conditions for Euler circuits and Paths):-

Statement

A connected multigraph with at least two vertices has an Euler circuit iff each of its vertices has even degree (1)

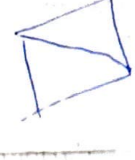
a – b – c – d – a

(2) A connected multigraph has an Euler path but not an Euler circuit iff it has exactly two vertices of odd degree. 1



a & d has odd degree

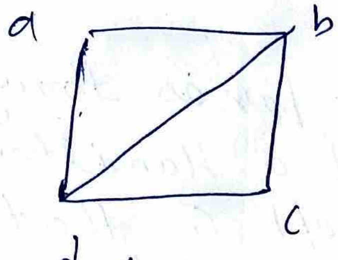
Example (2) of above

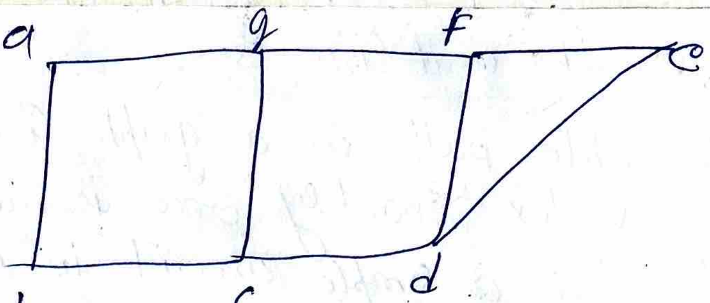


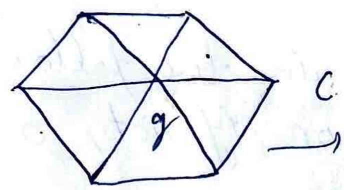
→ euler path but not EC

directly then this has Euler path but not euler circuit

Excercise ....







→ G_1 contains exactly two vertices of odd degree namely b & d. Hence, it has an Euler path that must have b and d as its endpoints.

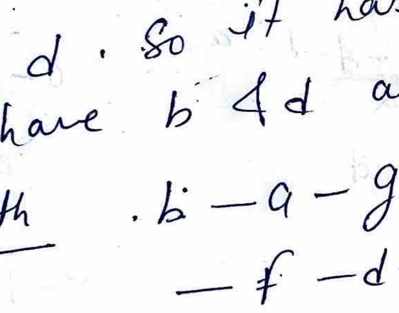
Euler path ⟶ d — a — b — c — d — b.

(2) G_2 has exactly two vertices of odd degree namely b and d. So it has has Euler path that must have b & d as its endpoints.

Euler path . b — a — g — f — e — d — c — g — b — c — f — d.

(3) G_3 has no Euler path because it has six vertices of odd degree.

(2) Example



$a \overset{e_1}{—} b \overset{e_7}{—} e \overset{e_5}{—} a \overset{e_4}{—} c \overset{e_9}{—} e \overset{e_6}{—} d$

$\overset{e_2}{—} b \overset{e_8}{—} e \overset{e_7}{—} c \overset{e_3}{—} d$

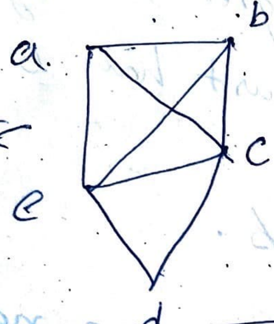
Euler path every edge cover and no edge repeat

but not have Euler circuit

Hamilton Paths and Circuits: -

Defn:- A simple path in a graph G that passes through every vertex exactly once is called a Hamilton path, and a simple circuit in a graph G that passes through every vertex exactly once is called a Hamilton circuit.

Ques Which of the simple graph in the following graph have a Hamilton circuit, or if not, a Hamilton paths?



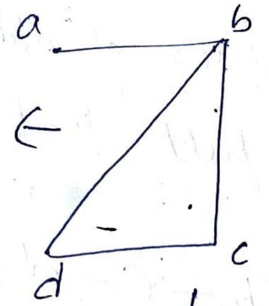
Has HC

a, b, c, d, e, a

a - b - c - d - e - a

Hamilton circuit

& HP



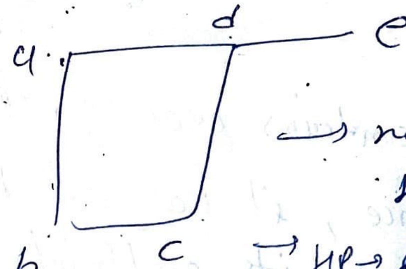
Not have a HP nor HC

~~a - b - c - d - b - a~~

a - b - d

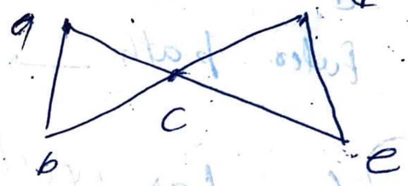
5

$\frac{5}{2} = 2.5$



→ not have H.C

→ HP → a - b - c - d - e



a - c - d - e - c

not have HC & HP

a - b - c - d - e

Note K_n for $n \geq 3$ has a HC.

Dirac's Theorem:- If G is a simple graph with n vertices with $n \geq 3$ such that the degree of every vertex in G is at least $n/2$ then G has a Hamilton circuit.

(1) HC

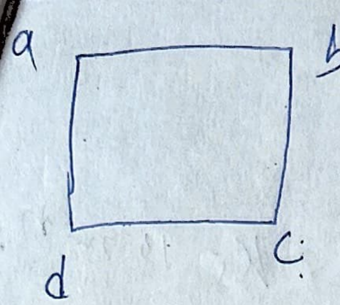
(1) Ex $n = 6$, and each vertex has $\frac{6}{2} = 3$ degree then G has a HC

$n = 4$

$\frac{n}{2} = \frac{4}{2} = 2$

ORe's Theorem:- If G is a simple graph with n vertices with $n \geq 3$ such that $\deg(u) + \deg(v) \geq n$ for every pair of non adjacent vertices u and v in G, then G has a Hamilton circuit.

→ Applications for networking, path or circuit that visits each road intersection



→ Dirac's Theorem Application

$n = 4 \geq 3$

and deg of very vertex is at least $\frac{n}{2} = \frac{4}{2} = 2$

(i.e more that $\frac{n}{2}$ is possible)

$\Rightarrow$ G has H C.

Ore's Theorem application ← Let a & c and b and d are non adjacent in G

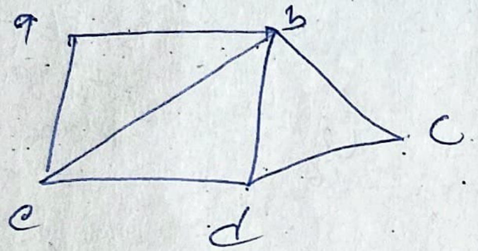
& $\deg(a) = 2 = \deg(b) = \deg(c) = \deg(d)$

such that $\deg(a) + \deg(c) = 2+2 = 4$ (≥ 4 → no. of vertices)

& $\deg(b) + \deg(d) = 2+2 = 4$

$\Rightarrow$ G has a Hamilton circuit

Qus 2



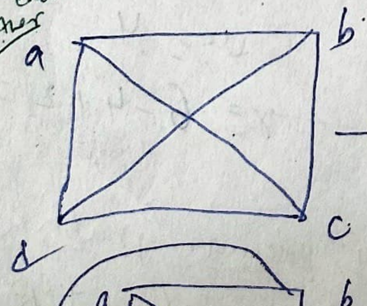
→ Hamilton Path exists or not

$\Rightarrow$ NO

Planar Graphs:- A graph is called planar if it can be drawn in the plane (i.e in 2D) without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a planar representation of the graph → circuit designing (Application) civil engineering

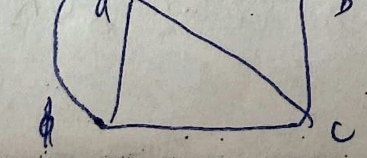
in such a way that no edges cross each other

Eg:-

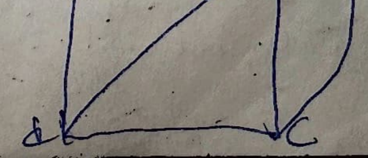


→ K_3 → planar or not

can we draw this graph without crossing any edges?

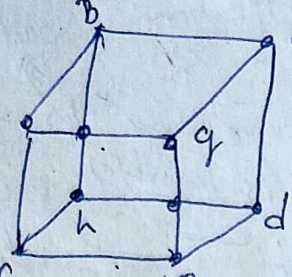


or

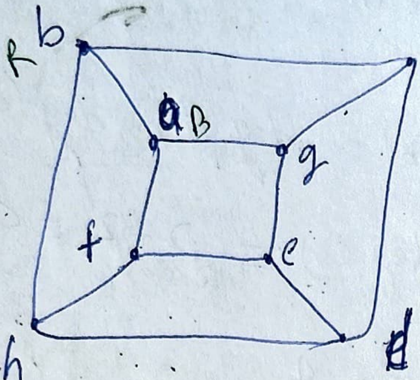


$\Rightarrow$ planar

②

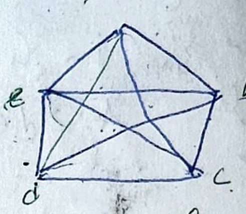


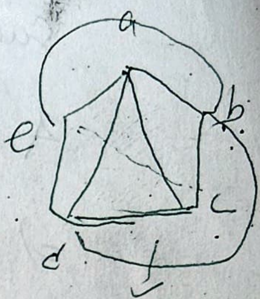
→ check planar or not



→ planar

→ K_5 is not p





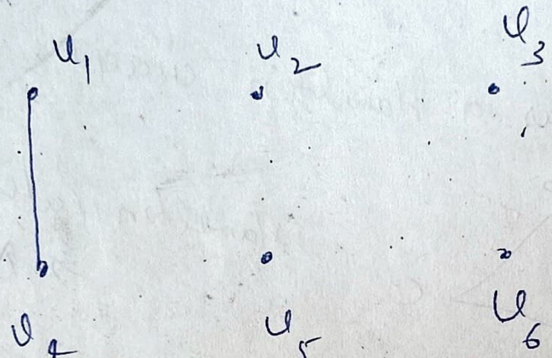
we can't draw this graph in a plane without crossing edge → non planar

$K_{3,3}$

↓

non planar

check



Applications of PG → design of electronic circuits.

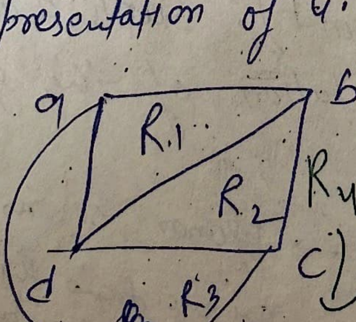
Imp **Euler's formula:** — → A P.G divides the plane into regions including ~~bounded~~ unbounded regions

Th^m: Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

Regions ~~means~~

↓

area enclosed between two vertices



$e = 6$

$v = 4$

$r = 6 - 4 + 2 = 2 + 2 = 4$

R_1, R_2, R_3 → bounded

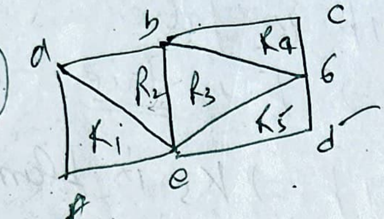
R_4 → unbounded

] in every planar graph we always including UR also.

Suppose that a connected planar ~~gr~~ simple graph has 15 vertices, each of degree 2. Into how many regions does a representation of this planar graph split the plane?

Sol^n:- No. of vertices, $v = 15$ and each has degree 2

i.e total no. of degree of vertices = sum of ~~vertices~~ degree of vertices



$$= \sum_{v \in V} \deg(v)$$

$$= 15 \times 2 = 30$$

By using the theorem,

if G be UG with m edges then $2m = \sum_{v \in V} \deg(v)$

$$2m = 30$$

$$\Rightarrow m = 15 = \text{no. of edges} = e$$

By using Euler's formula,

no. of regions $r = e - v + 2$

$$= 15 - 15 + 2$$

$$= 2$$

solve for $v = 20$ with each of degree 3.

$$2m = 20 \times 3$$

$$m = 30$$

$$r = 30 - 20 + 2 = 12$$

(1) Corollary (Inequality) If G is a connected planar simple graph with e edges and v vertices, where $v \geq 3$, then

$$\boxed{e \leq 3v - 6}$$

(2) If G is a connected planar simple graph, then G has a vertex of degree not exceeding five.

Qus Show that K_5 is nonplanar.

Soln: Let K_5 is planar

no. of vertices $|v| = 5$, $|e| = 10$

By using above corollary

$e \leq 3v - 6$

By using ~~contra~~ ~~tion~~ proof of contradiction or

$\cancel{10 \leq 3 \cdot 15 - 6 = 45 - 6 = 39}$

$10 \leq 3 \cdot 5 - 6 = 15 - 6 = 9$

$\neg p$ is false

$\Rightarrow p$ is true

$10 \leq 9$

$\Rightarrow K_5$ is not planar

$\Rightarrow K_5$ is ~~be~~ nonplanar

$\neg p \Rightarrow K_5$ is planar

so we can apply the ~~corollary~~ corollary ①

② Corollary :- If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuits of length three, then $e \leq 2v - 4$

Qus Show that $K_{3,3}$ is nonplanar. (using above corollary)

$K_{3,3}$ has no circuits of length 3 (becas it is bipartite)

Let $K_{3,3}$ is planar, $|v| = 6$, $|e| = 9$

the $e \leq 2v - 4 \Rightarrow 9 \leq 2 \cdot 6 - 4 = 12 - 8 = 8$

$9 \leq 8$

$\Rightarrow K_{3,3}$ is not planar

$\Rightarrow K_{3,3}$ is ~~pl~~ nonplanar

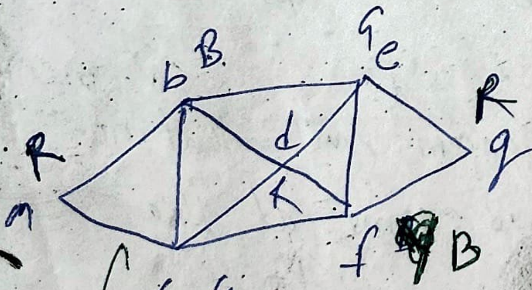
...ph coloring :-

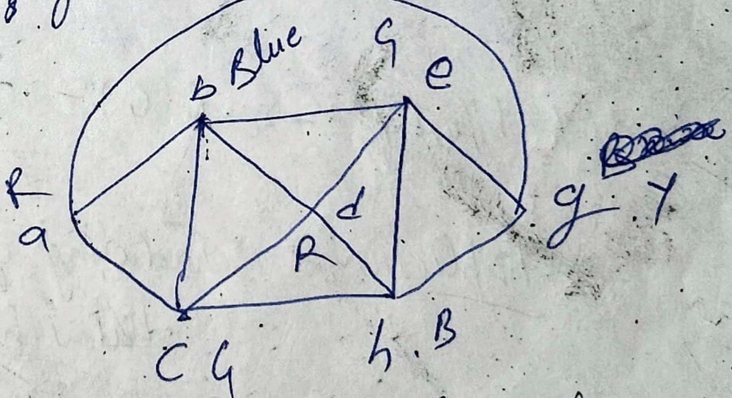
Def^n :- A coloring of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

Chromatic Number:- The chromatic number of a graph is the least number of colors needed for a coloring of this graph. The chromatic number of a graph G is denoted by $\chi(G)$. { The graph is called K-chromatic

Theorem (The four color theorem):- The chromatic number of a planar graph is no greater than four.

Ques: What are the chromatic number of the graph G & H of the following graphs.



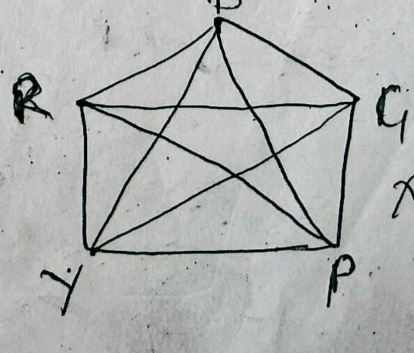


R → Red
B → Blue
G → Green
Y → Yellow

$\chi(G) = 3$

$\chi(G) = 4$

K_5 → non-planar graph



$\chi(G) = 5$

Chromatic number of Special Graphs:-

1. Complete, $K_n \rightarrow \chi(K_n) = n$ i.e chromatic number of com graph = no. of vertices

2. Cycle, C_n, for $n \geq 3$ $\chi(C_n) = 2$, n is even integer with n
   $\chi(C_n) = 3$, n is odd integer

3. Regular

4. wheel $\rightarrow$ $n \geq 3$ when n is even, $\chi(W_n) = 3$
   when n is odd, $\chi(W_n) = 4$

5. Cube

6. bipartite $\rightarrow \chi(G) = 2$

7. complete bipartite $\rightarrow \chi(G) = 2$ or $\chi(K_{m,n}) = 2$

Application $\rightarrow$ Scheduling your exam such that no student has two exams at the same time.

Graph and its properties

Introduction of Rooted tree and its properties, m-ary and full m-ary tree

Spanning tree and its properties, minimum spanning tree - Prims and Kruskal Algorithm

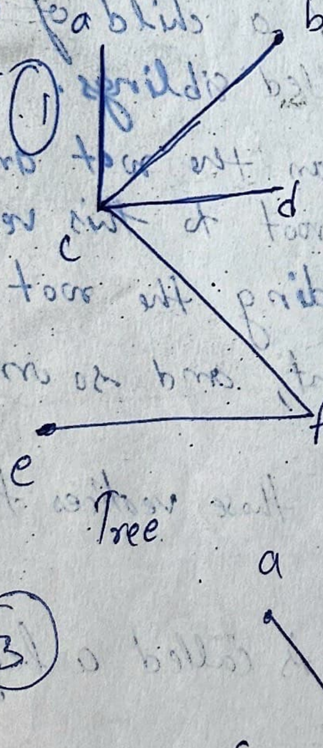
(4) Infix, prefix and post fix notation.

Tree :- A connected (UG graph) that contains no simple circuits is called a tree. (A connected graph without any circuit).

Applications → Computer science in data structure

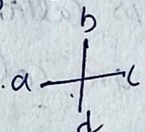
Note :- A tree can't contain multiple edges or loops.

Ex :- (1)

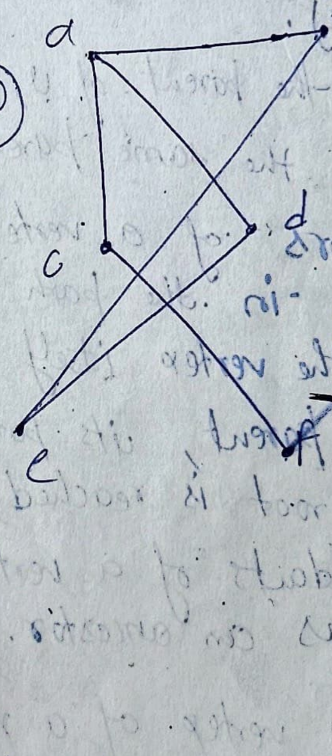


Tree

(2)



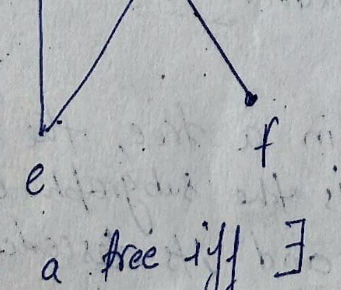
(2)



b - e - d - a - b → Simple circuit

not a tree (because it has a simple circuit)

(3)



→ not a tree bcz it is not connected

no path betⁿ a to c or a to d

Thm :- An UG is a tree iff ∃ a unique simple path between any two of its vertices

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

indicated

→ example on next page.

*[A particular vertex of a tree is indicated as the root.]

(or named)

When we specify a root, we assign a direction to each edge. and ∃ a unique path from the root to each vertex of the graph).]

Suppose T is a rooted tree and v is a vertex in T (other than the root), *the parent of v is the unique vertex u such that there is a directed edge from u to v. (Such vertex is unique).

* When u is the parent of v, v is called a child of u.

* Vertices with the same parent are called siblings.

* The ancestors of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root (i.e., its parent, its parent's parent, and so on, until the root is reached).

previous generation

* The descendants of a vertex v are those vertices that have v as an ancestor.

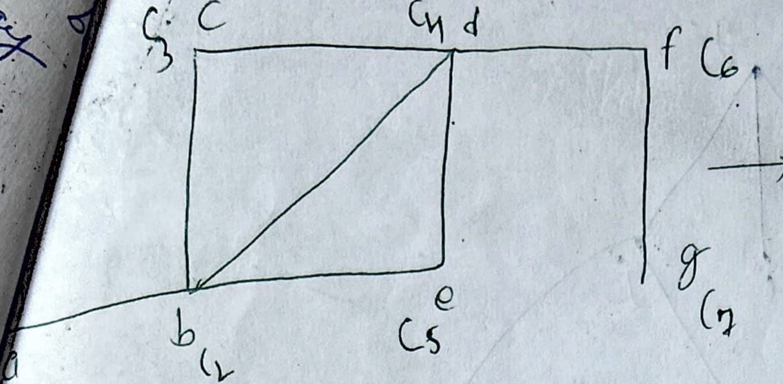
next generation

* **Leaf:-** A vertex of a rooted tree is called a leaf if it has no children.

* Vertices that have children are called internal vertices.

No Need

Subtree:- If a is a vertex in a tree, the subtree with a as its root is the subgraph of the tree consisting of a and its descendants and all edges incident to these descendants.

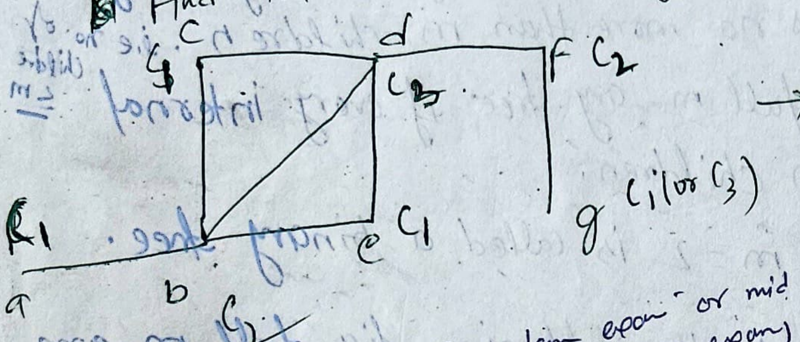


assign this graph vertex by color but try to use minimum color (i.e our spend will be less to use color)

Idea behind Chromatic Number

Coloring the graph by using minimum number of colors such that no two adjacent vertex has same color

Find the choromatic no. of this graph → to set your time table or exam schedule such no classes are clash

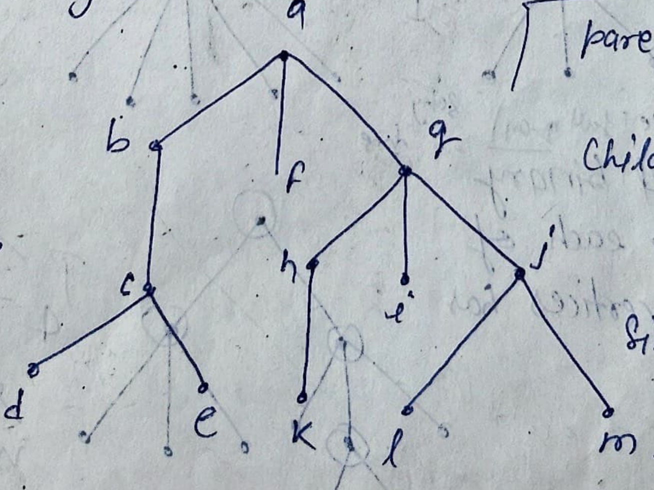


(End term exam or mid term exam)

→ so minimum no. of color in this graph is use 3

i.e Chromatic no. of this graph = 3.

Qus: Find the parent of c, the children of g, the siblings of h, all ancestors of e, all descendants of b, all internal vertices, and all leaves of the following rooted tree T.



T →

The descendants of b are c, d & e.

The internal vertices are a, b, c, g, h and j.

Leave are d, e, f, i, k, l and m.

This a rooted tree T.

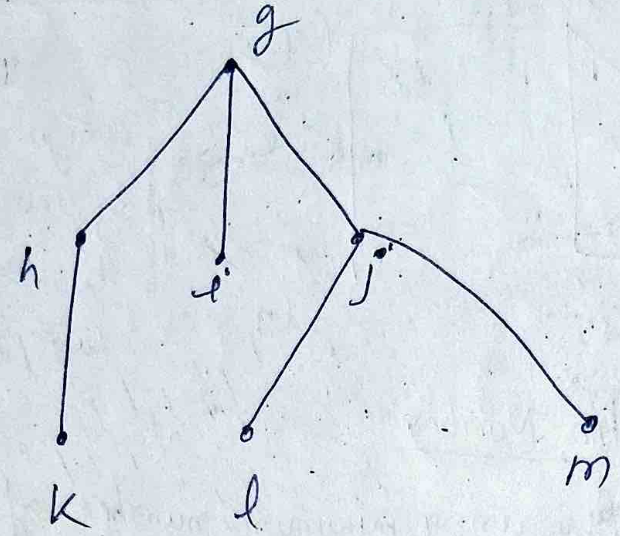
Subtree rooted at g is shown next fig

parent of c is b.

Children of g are h, i & j.

Siblings of h are i and j.

The ancestors of e are c, b and a.



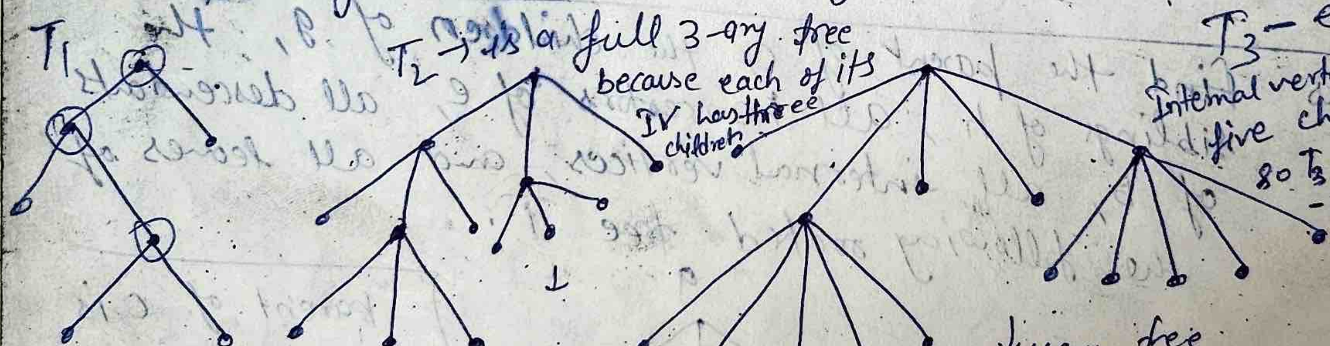
The subtree Rooted at g.

Defn:- A rooted tree is called an m-ary tree if every internal vertex has no more than m children. i.e no. of children $\leq m$

The tree is called a full m-ary tree if every internal vertex has exactly m children.

An m-ary tree with $m = 2$ is called a binary tree.

Ex2:- Are the rooted trees in the following fig full m-ary trees for some positive integer m?



T_2 → is a full 3-ary tree because each of its IV has three children

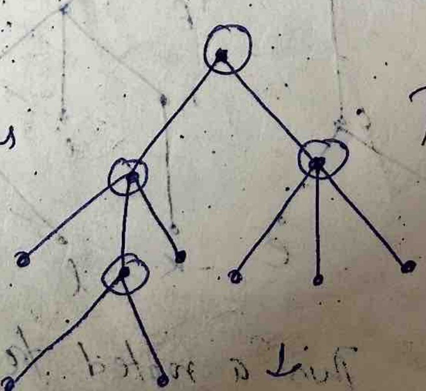
T_3 — each internal vertex has five children, so T_3 is a full -5 ary tree

4ary tree

2-ary tree (full m-ary)

3ary tree

T_1 is a full binary tree because each of its internal vertices has children.



T_4 → is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children.

3ary bcz no internal vertex has more than 3 children.

Theorem:- A tree with n vertices has $n-1$ edges.

Counting vertices in full m-ary trees

Thm:- A full m-ary tree with i internal vertices contains $n = mi + 1$ vertices.

Thm:- A full m-ary tree with

i) n vertices has $i = (n-1)/m$ internal vertices and
$l = [(m-1)n+1]/m$ leaves,

ii) i internal vertices has $n = mi + 1$ vertices and
$l = (m-1)i + 1$ leaves,

iii) l leaves has $n = (ml-1)/(m-1)$ vertices and
$i = (l-1)/(m-1)$ internal vertices.

Th_2^m:- There are at most m^h leaves in an m-ary tree of height h.

* A complete m-ary tree is a full m-ary tree in which every leaf is at the same level.

$$\begin{array}{r} 45 \\ 9 \\ \hline 4050 \end{array}$$

Infix, Prefix, and Postfix Notation:- ~~There are three types of representation are used to~~ write a mathematical expression (A mathematical expressions can be represented by a tree).

Infix form:- operation between any two numbers/variables ~~such that~~ called infix mathematical expression that use $a * b$, a/b, $a \pounds b$. $x + y/x + 3$, $(x + (y/x)) + 3$

Prefix & Postfix ~~boad~~ mass rule. Boadmas rule → apply.

Prefix form:- [We obtain the prefix form of an expression when we traverse its rooted tree in preorder. (), *, /, +, -

* Expressions written in prefix form are said to be in Polish notation.] infix $a*b$ → prefix $*ab$ (go left to right)

Ques:- What is the prefix for $((x+y) \uparrow 2) + ((x-4)/3)$?

Operations that used before the variables/numbers called prefix.

Sol^n:- We obtain the prefix form for this expression by traversing the binary tree that represents it in preorder. This produces $+ \uparrow + x\ y\ 2 / - x\ 4$ $*$ prefix form

Ex:- ~~f~~ $+ \uparrow + x y 2 / - x 4 3$ (for tree)

Postfix:- Operations that used ~~before~~ after the variables/nos called posfix.

Ex. for same expression $x y + \uparrow x 4 - 3 / +$.

Ex 2: Priorities ( )

both have same priority / or $*$ (solve from left to right when we have more than one / or $*$).

" + or - (")

Sol^n 2 Prefix $((x+y)\uparrow 2) + ((x-4)/3)$

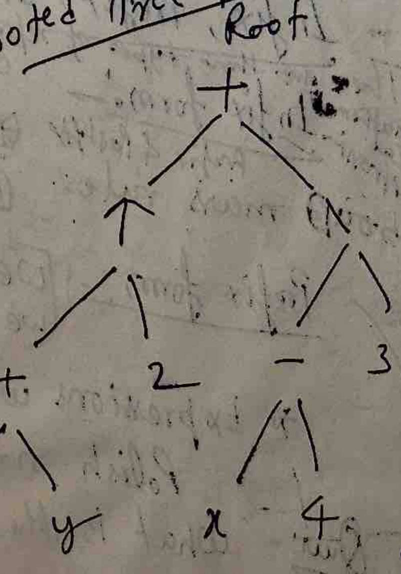
~~$(+xy \uparrow 2) + ((x-4)/3)$~~

~~$\uparrow + xy 2 + ((x-4)/3)$~~ $(-x4/3)$

~~$\uparrow + xy + / - x43$~~ Rooted Tree presentation

$+ \underbrace{\cancel{\uparrow + xy}}_{a} \underbrace{\cancel{/ - x43}}_{b}$

Root



$\uparrow (x+y) 2 + ((x-4)/3)$

$\uparrow (x+y) 2 + / (x-4) 3$

$\uparrow + xy 2 + / - x 4 3$

$+ \uparrow + xy 2 / - x 4 3$

fix form :- We obtain the postfix form of an expression by traversing its binary tree in postorder.

* Expressions written in postfix form are said to be in reverse polish notation. → i.e $a * b = ab*$

Ques:- What is the postfix form of the expression $((x+y) \uparrow 2) + ((x-4)/3)$?

Soln:- The postfix form of the expression is obtained by carrying out a postorder traversal of the binary tree for this expression, shown in the following fig

This produces the posfix expression : $x\,y + 2 \uparrow x\,4 - 3 / +$

Soln:- Postfix :-

$$((x+y) \uparrow 2) + ((x-4)/3)$$

$$(\cancel{xy+\uparrow 2}) + (\cancel{x4 - /3})$$

$$\cancel{xy+\uparrow}$$

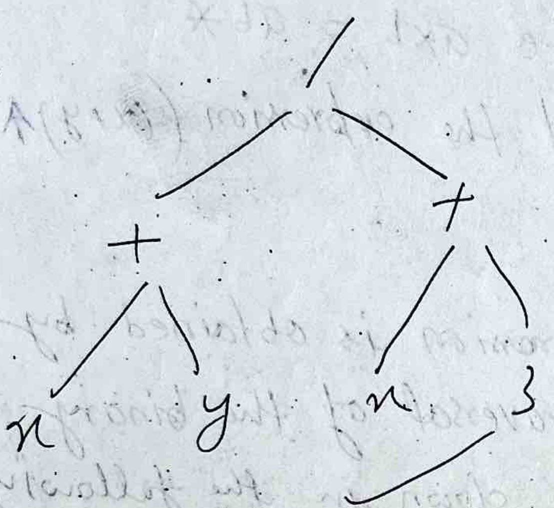
$$(x+y)\,2 \uparrow + (x-4)\,3\,/$$

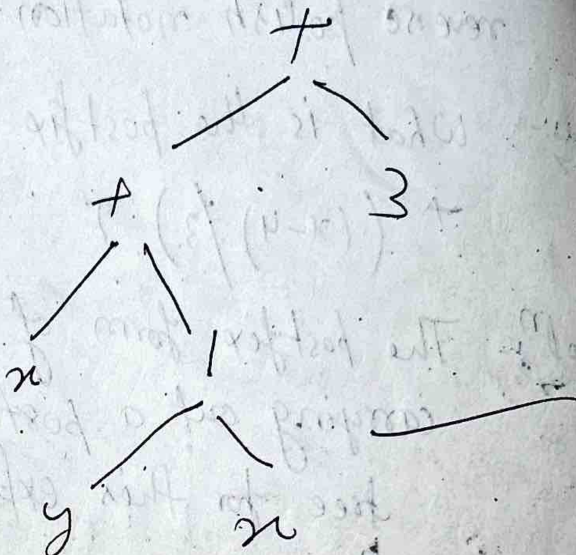
$$xy+2 \uparrow + x4-3/$$

$$xy+2 \uparrow x4-3/+$$

Rooted tree Representation

① $(x+y) / (x+3)$ ② $(x+(y/x))+3$





Rooted tree of the above expression.

CLASS TEST SHEET

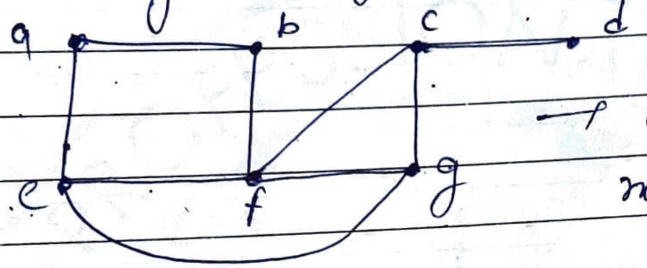
Roll No. Reg. No. 12/80 Course Code Date of Test Section Test No.

Spanning Trees

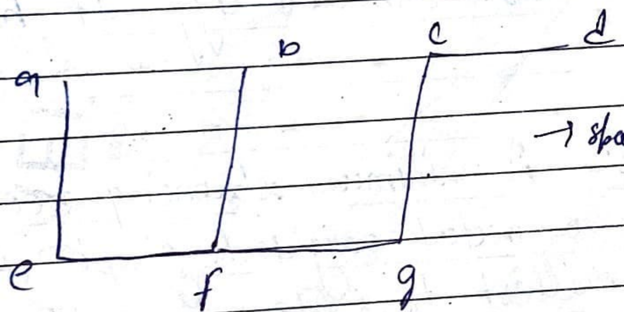
Spanning Trees :- Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

Ques Find a spanning tree of the following simple graph G.

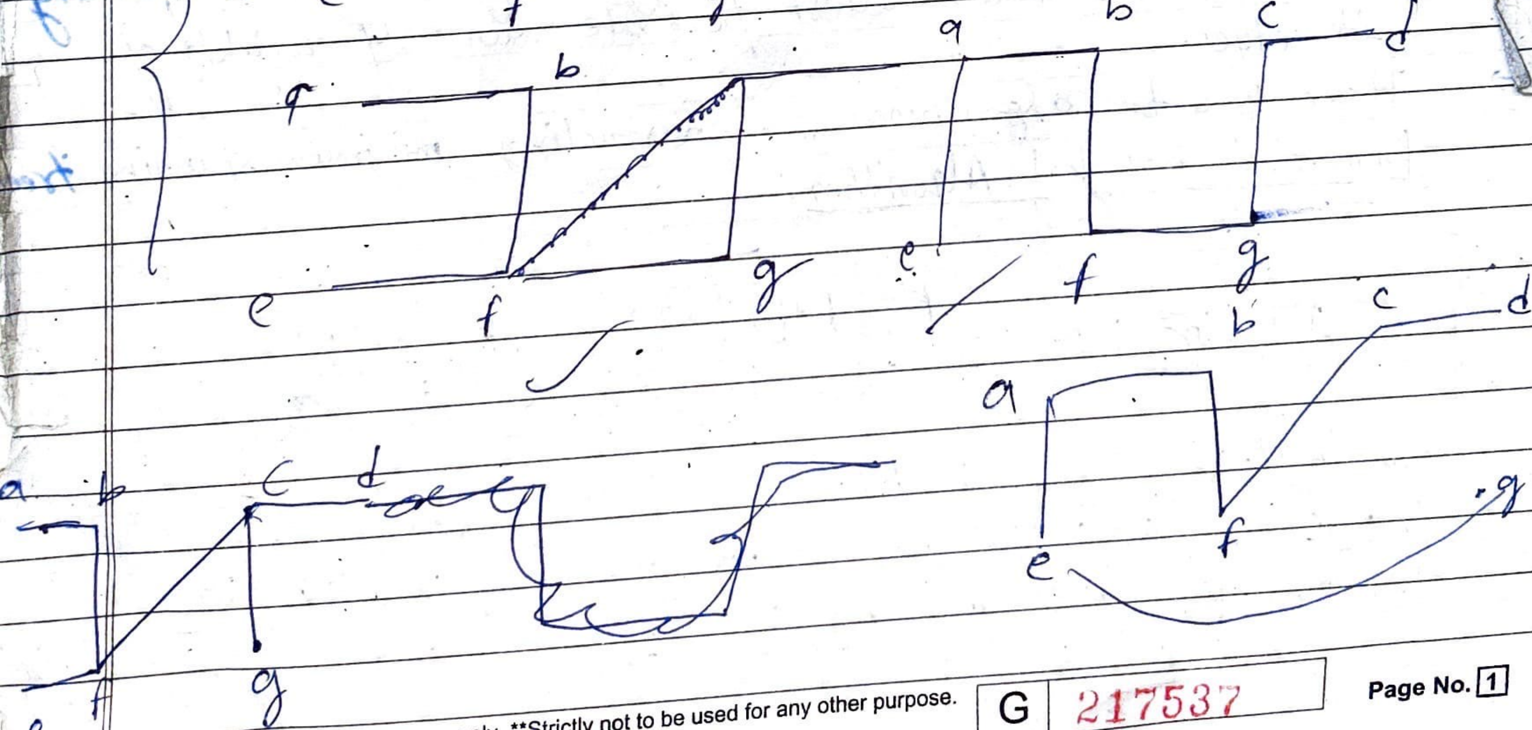
Soln :-



→ Graph is connected but it's not a tree because it contains simple circuits.



→ spanning tree.



*To be used for class test purpose only **Strictly not to be used for any other purpose.

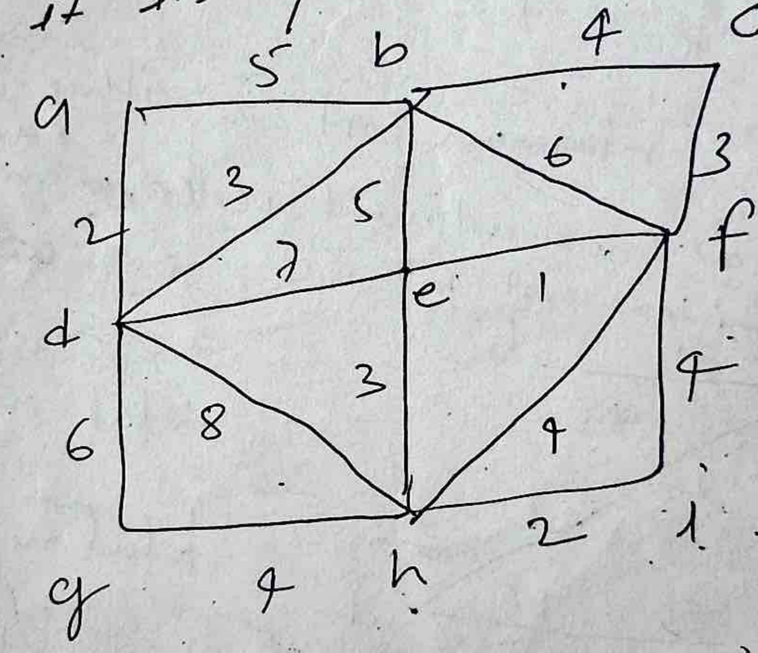
G 217537

Kruskal's Algorithm :- In this method first we select the minimum cost edge. Again select the minimum cost edge from the remaining part of the graph such that circuit should not construct. Repeat the process till the number of edges becomes $(n-1)$

Note :- Total number of edges in minimum spanning tree is $(n-1)$.

Both algorithm have ~~both~~ sam MST cost

Now verify ~~that~~ with Prim's and Kruskal's algorithm that both algorithms yield the same MST cost and it is equal to 22.

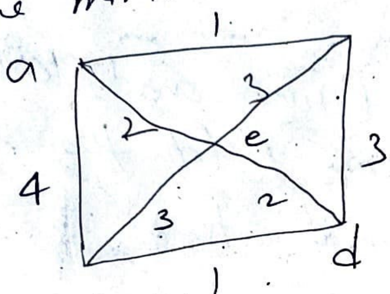


Prim's Algorithm :- Prim's algorithm is a method to find the minimum spanning tree cost. In this method, first select the minimum cost edge. Next select that edge which is adjacent to at least one of the vertex of all selected edges but has minimum cost such that circuit does not becomes. Repeat the process till the number of edges becomes $(n-1)$.

$n \to$ no. of vertex

MST cost $\to$ minimum spanning tree cost

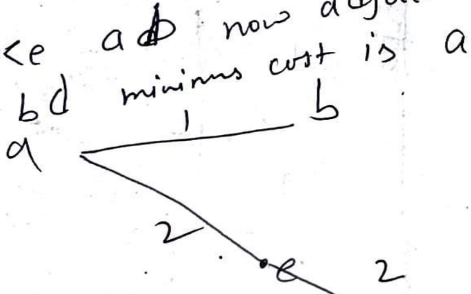
Qus 2 Find the minimum cost.



1st Method

By using prim's minimum cost is either take ab or cd

let take ab now adjacent vertexcies of a are ae & ac & b are be, bd minimum cost is ac, now ~~cd~~ AV of e are ce & cd.



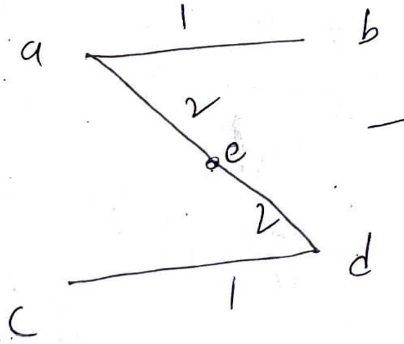
spanning tree

total no. of edges = 4
$(5-1)$

total no. of cost is
$1+2+2+1 = 6$

2nd Method

Krustkal's —



same

i.e both method gave same MST cost