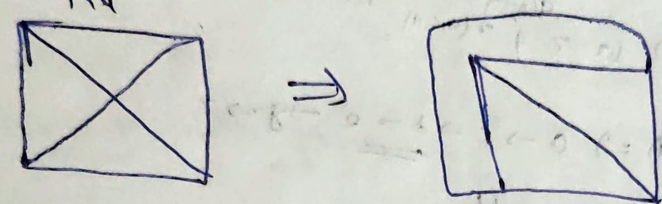
Unit - 5

Planner Graph

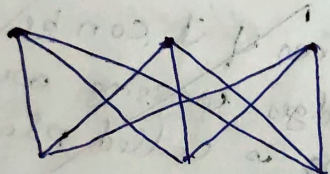
* A graph is planner if it can be drawn in the plane without any edges crossing or intersecting each other. Such a drawing is called planner representation of graph.

K_4



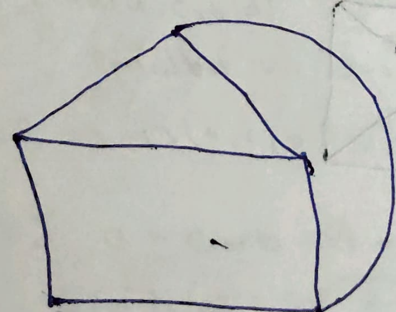
$K_{3,3}$

K_5



Not Planner.

K_5



Non planner

Euler's formula :-

* Let G be a connected planar graph with e-edges & v-vertices. Let r be a no. of region in a planar representation of G.

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

Q) Let a connected planar graph has 20 vertices, each of degree 3. Into how many regions does a representation of this planar graph splits the plane.

Ans. Given $v = 20$ vertices.

By handshaking theorem

$2e = 20 \times 3$

$\underline{e = 30}$

So ; $r = e - v + 2$

$r = 30 - 20 + 2 \Rightarrow \underline{12}$

Q) $V = 12$, 8 having degree 2, & 4 vertices having degree 4. Find the no. of region.

Ans $2e = 8 \times 2 + 4 \times 4$.

$\Rightarrow 16 + 16 = 32 = 2e$

$\boxed{e = 16}$

So, $r = e - v + 2$

$\Rightarrow 16 - 12 + 2$

$\Rightarrow \underline{6}$

Important Results:-

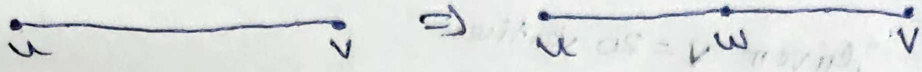
* If G is a connected planar simple graph with e-edges & v-vertices, where $v \geq 3$.

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

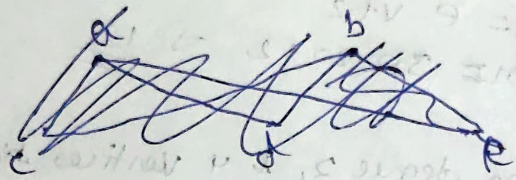
* If G is connected planar graph, then G has a vertex of degree not exceeding five.

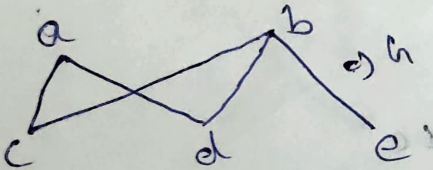
* If G is a connected planar simple graph with e-edges and v-vertices, where $v \geq 3$ & no. of circuit of length 3 then $\boxed{e \leq 2v - 4}$

* If G is a plannar graph, so will be any graph obtained by removing an edge $\{u,v\}$ & adding a new vertex w together with an edge $\{u,w\}$ & $\{w,v\}$. such an operation is called elementary subdivisions.

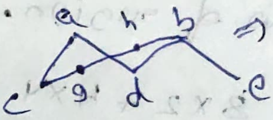


* The graph $G_1=(V_1,E_1)$ & $G_2=(V_2,E_2)$ are called homomorphic if they are obtained from the same graph by a seq. of elementary subdivisions.

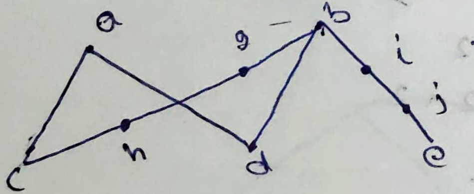




=> G



=> G_1



=> G_2

Karatowski's Theorem

* A graph is non-plannar iff it contains a subgraph homomorphic to $K_{3,3}$ or K_5.

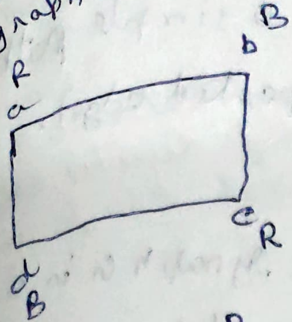
Graph Colouring:-

* Assigning colour to the vertices so that no two adjacent vertices get same colour.

Chromatic Number.

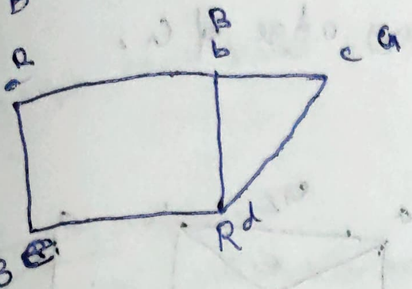
It is the least no. of colours needed for colouring the graph.

Ex



chromatic no :- 2

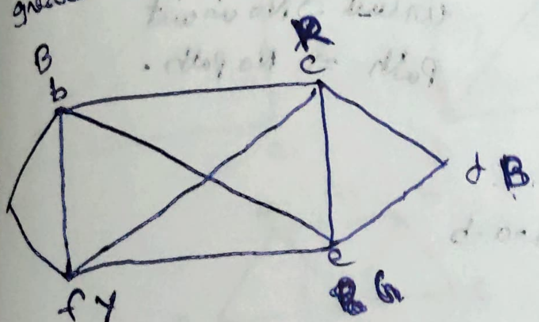
Ex



chromatic no :- 3

colour theorem:-

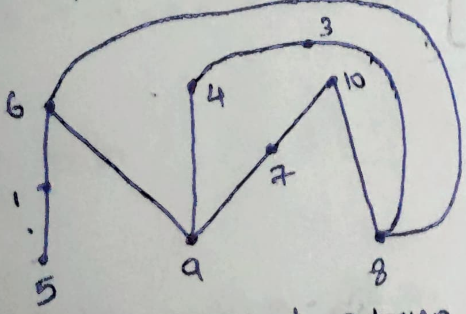
* The chromatic no. of a plannar graph is not greater than 4.



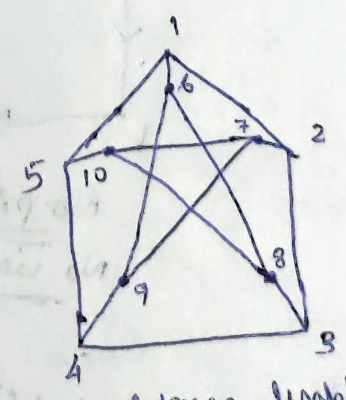
chromatic no. = 4

Theorem:-

* A graph is non-planar iff it contains a subgraph homomorphic to $K_{3,3}$ or K_5.



a subgraph of Peterson graph.



Petersen graph.

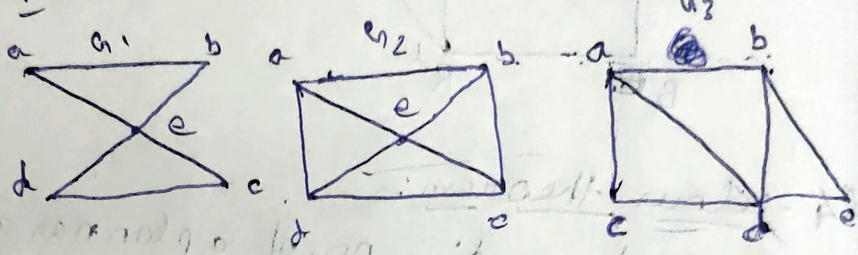
Euler Path:-

* An Euler Path in G is a simple path containing every edge of G (no repeated edge).

Euler Circuit:-

* An euler circuit in a graph G is a simple circuit containing every edge of G.

* Example:-

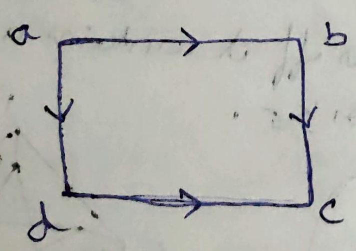


G1
Circuit ⇒ a-b-e-c-d-e-a
Path ⇒ No path.

G2
Circuit ⇒ No circuit
Path ⇒ No path.

G3
Circuit ⇒ No circuit
Path ⇒ a-c-d-e-b-d-a-b

* Example:-



No path
No circuit

Necessary & sufficient condition for Euler circuit & path

* Theorem:-
A connected multigraph with atleast two

vertices has an Euler circuit if and only if each of its vertices has even degree.

* Theorem :-

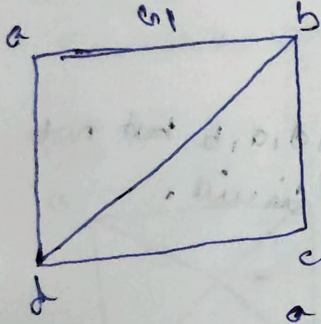
A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree.

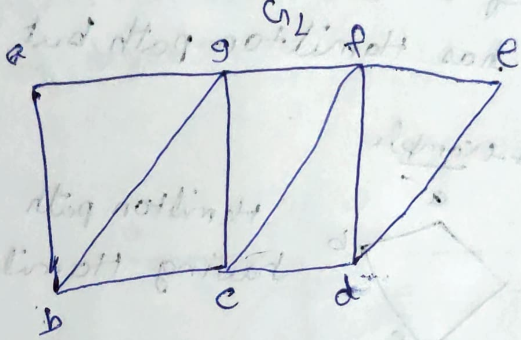
Note:-

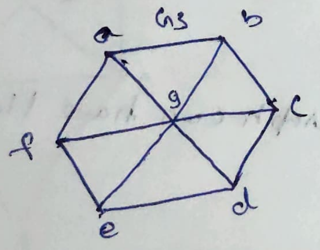
* A graph cant have both Euler's path and Euler's circuit.

Eulerian graph :-

* A graph which contains Euler circuit is called Eulerian graph.







G1

d - a - b - c - d - b (Euler Path)

G2

Euler Path

G3

No Euler path as well as Euler circuit.

Hamilton Path

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

Hamilton Circuit:-

* A simple circuit in a graph G that passes through every vertex exactly once is called Hamilton circuit.

Hamiltonian Graph

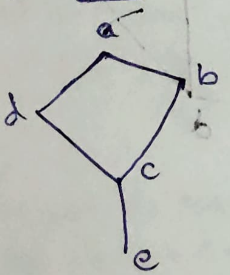
* A graph contains Hamiltonian circuit.

Important pts:-

* A graph can have both Hamilton path as well as Hamilton circuit.

* If a graph has Hamilton circuit then it also has Hamilton path but converse is not true.

* Example



Hamilton path e, c, d, a, b but not having Hamilton circuit.

* only a connected graph can have Hamilton circuit / path.

* A graph with a vertex of degree one cannot have a Hamilton circuit.

Conditions for the existence of Hamilton Circuit

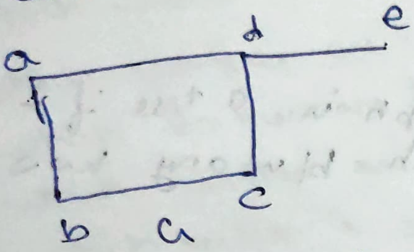
* Dirac theorem:-

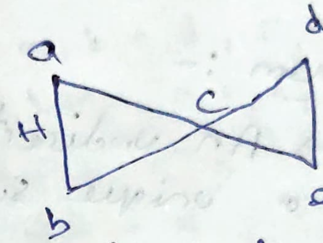
→ If G is a simple graph with n-vertices with $n \geq 3$ such that the degree of every vertex in G is atleast $\frac{n}{2}$, then G has a Hamilton circuit.

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 & v in G then G has a Hamilton circuit.

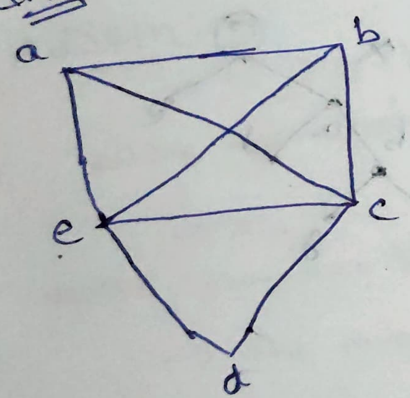
A Hamilton circuit cannot contain a smaller circuit within it.

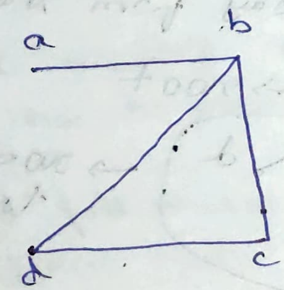


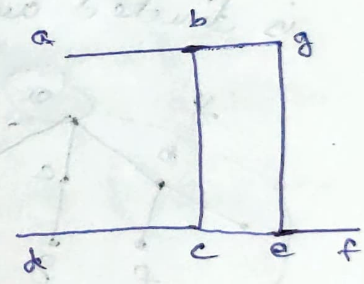


In G There is no Hamilton circuit in G because G has a vertex of degree zero.

In H:-



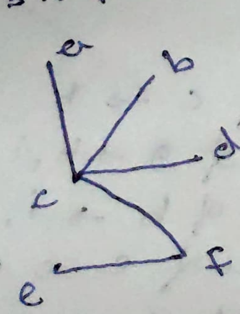


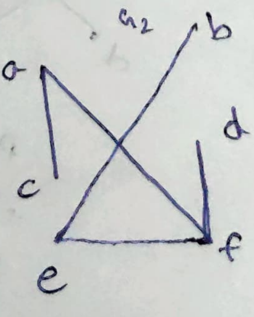


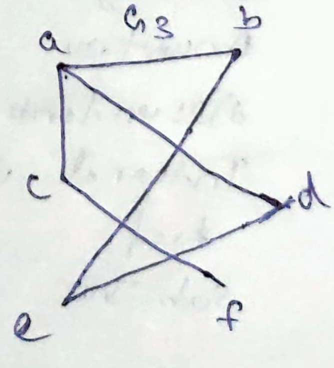
Tree graph and it properties

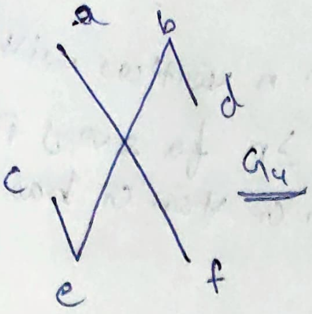
* Defn:-

⇒ A tree is a connected undirected graph with no simple circuit.









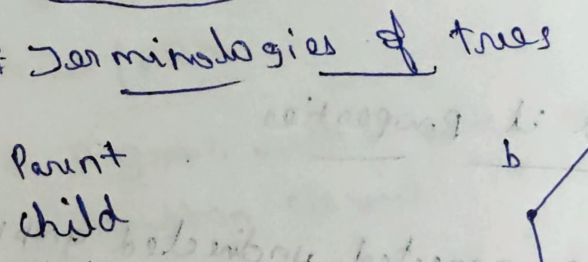
* Graphs containing no simple circuit that are not necessarily connected are forests.

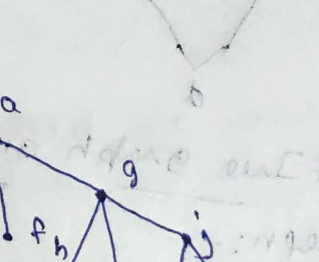
Theorem :-

→ An undirected graph is a tree if & only if there is a unique simple path b/w any two of its vertices.

* Defn:-

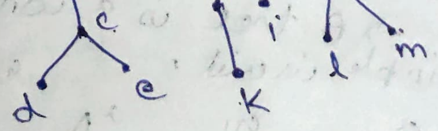
↔ A rooted trees is a tree in which one vertex has been designated as the root & every edge is directed away from the root.



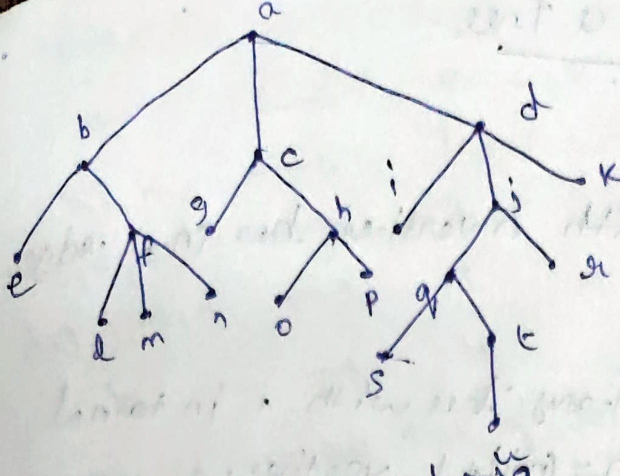


Terminologies of trees

Parent
child
siblings
Ancestors
descendants
Internal Vertices
leaf
Sub Tree



Parent of c → b
children of g → h, i, j
sibling of h → i, j
Ancestors of e → c, b, a
descendants of b → c, d, e
Internal vertices → a, b, c, g, h
Leaves → d, e, k, l, i, m.



which vertex is root → a.

which vertices are internal → a, b, c, d, f, h, j, q, t.

which vertices are leaves → e, l, m, n, o, p, s, u, r, k.

which vertices are children of j → q, r

which vertex is the parent of h → c

which vertex is the siblings of o → p

which vertices are ancestors of m → b, f, a

which vertices are descendent of b → e, f, l, m, n.

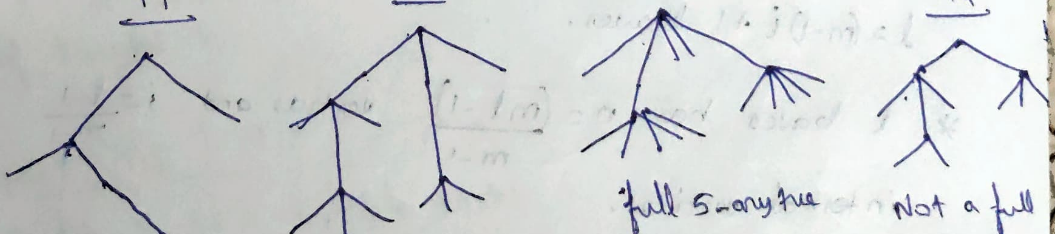
*Defn: –

→ A rooted tree is called m-ary tree if every internal vertex has no more than m children.

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

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

T_1 T_2 T_3 T_4



full binary tree

full 3 ary tree.

full 5-ary tree

Not a full m-ary tree for any m.

Properties of a tree

Theorem:-

* A tree with n vertices has $(n-1)$ edges.

Theorem

* A full m-ary tree with i internal vertices contains $n = (mi) + 1$ vertices.

Theorem:-

* A full m-ary tree with

→ n vertices has $i = \frac{(n-1)}{m}$ internal vertices

and $l = \frac{[(m-1)n+1]}{m}$ leaves.

Defn

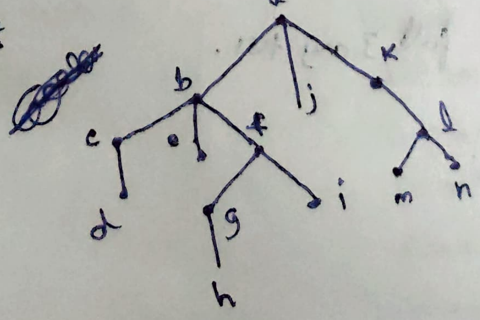
* The level of a vertex v in the rooted tree is the length of the unique path from the root to its vertex. The level of the root is defined to be zero. The height of a rooted tree is maximum of the levels of vertices.

* In other words, the height of a rooted tree is the length of the longest path from the root to any vertex.

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

* l leaves has $n = \frac{(ml-1)}{m-1}$ vertices and $i = \frac{l-1}{m-1}$ internal vertices.

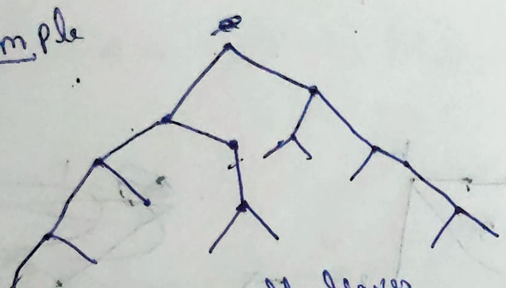
* Example



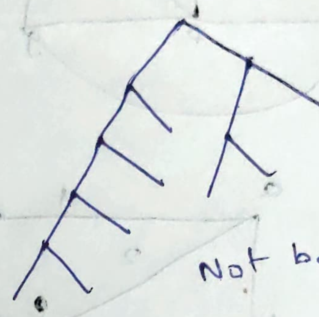
Vertex		Level
a	$\longrightarrow$	0
b, j, k	$\longrightarrow$	1
c, e, f l	$\longrightarrow$	2
d, g, i, m, n	$\longrightarrow$	3
h	$\longrightarrow$	4

* A rooted m-ary tree of height h is balanced if all leaves are at levels of h or $(h-1)$.

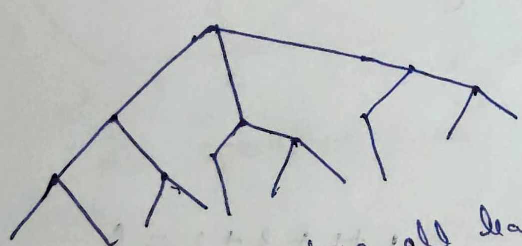
→ Example



Balanced since all leaves are at levels 3 & 4.



Not balanced.



Balanced since all leaves are at level 3...

Theorem :-

* There are almost m^n leaves in an m-ary tree of height h.

Spanning tree:-

Let G be a simple graph,

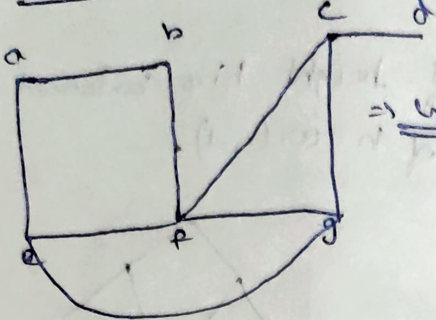
Defn * → A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.

* Theorem :-

→ A simple graph is connected if and only if it has a spanning tree.

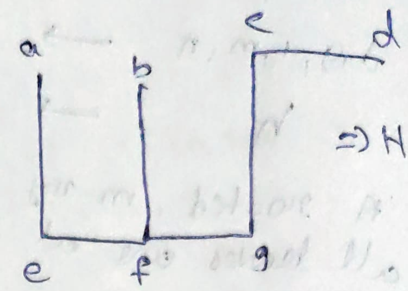
* A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edge.

* Examples :-



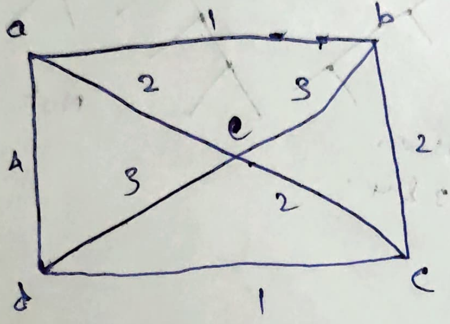
⇒ G

⇒

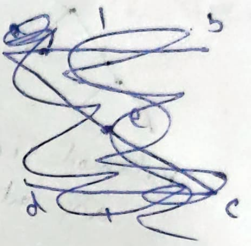


⇒ H

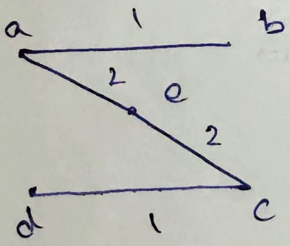
*



⇒



Prim's Algorithm :

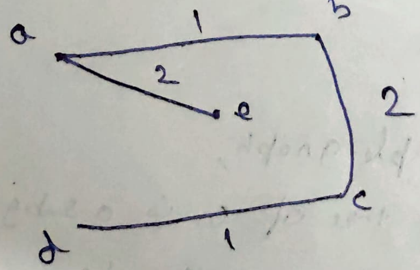


1 + 1 + 2 + 2 = 6

min

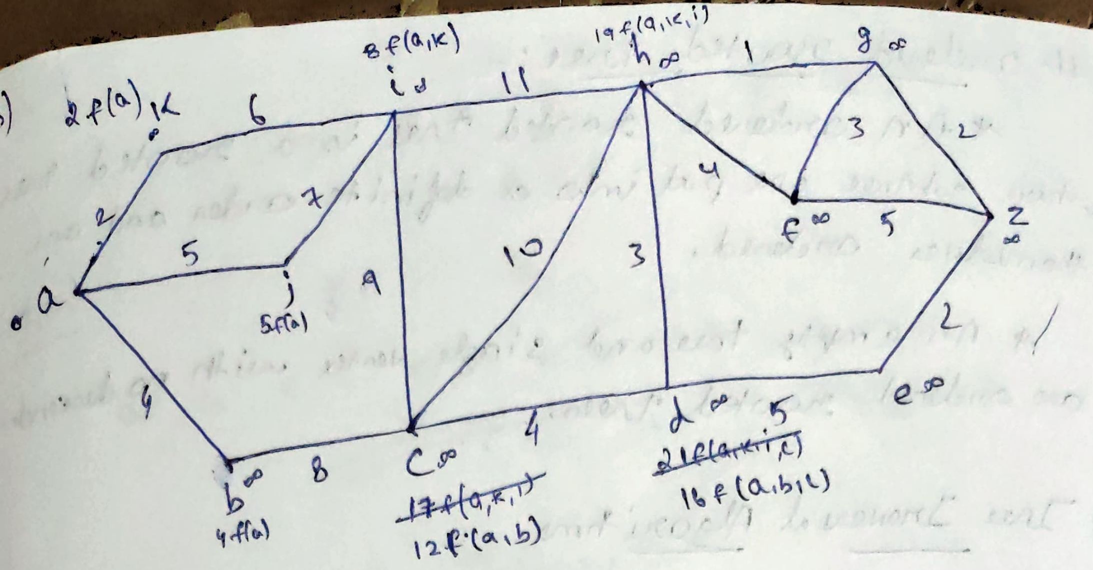
* Adjacent one.

Kruskal Algorithm :-

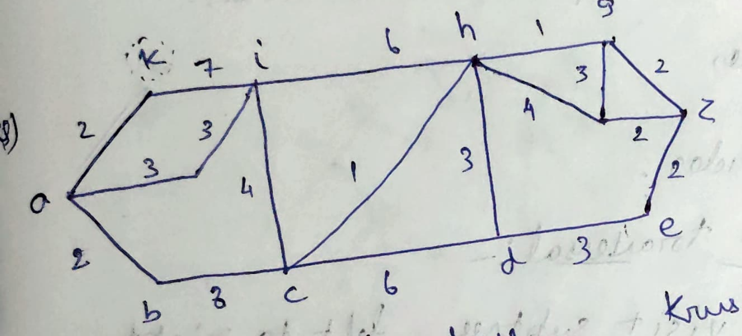


* Random one.

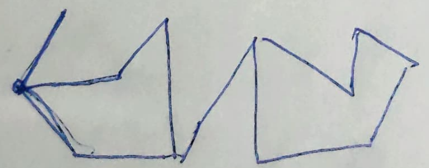
weight ⇒ 1+1+2+2 = 6



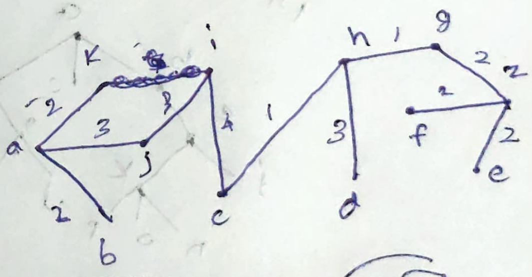
a, b, c, d, h, g, z => 22



Min spanning tree,



Krushal's Algorithm.



2+2+3+7+4+1+3+1+2+2

10+10+8 => 28+2

=> 30

Prime,

2+2+3+3+4+1+3+2+2+2+1

10 +10+5

=> 25

Ordered rooted tree:-

* An ordered rooted tree is a rooted tree whose subtree are put into a definite order and are themselves ordered.

* An empty tree and single vertex with no descendants are ordered rooted trees.

Tree Traversal Algorithms

* Procedures for systematically visiting every vertex of an ordered rooted tree are called traversal algorithm.

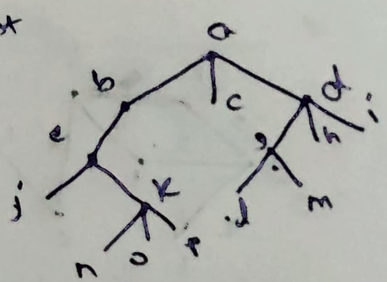
* Most commonly useful traversal order:-

→ Pre order
→ In order
→ Post order.

For Pre order traversal:-

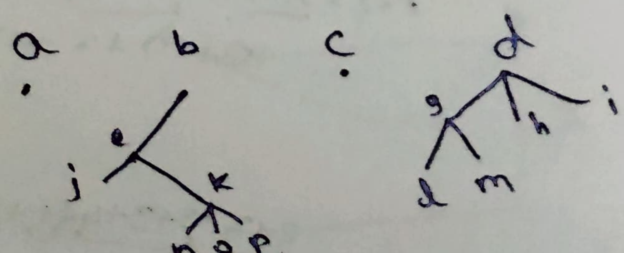
* Visit root, visit subtree, left to right.

1st



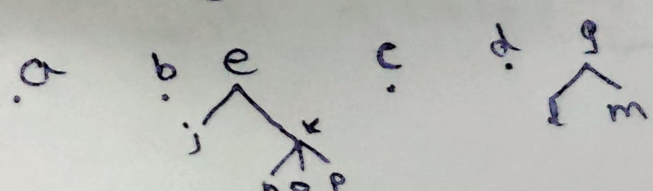
⇓

2nd



⇓

3rd



⇓

4th

a b e j 

c d g l m h i

⇓

5th

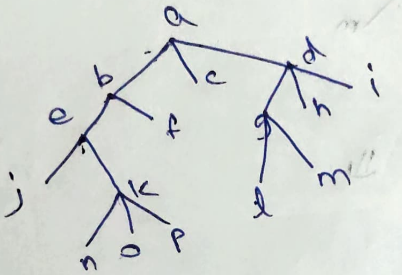
a b e j k n o p c d g l m h i

(Pre order)

For In Order Traversal

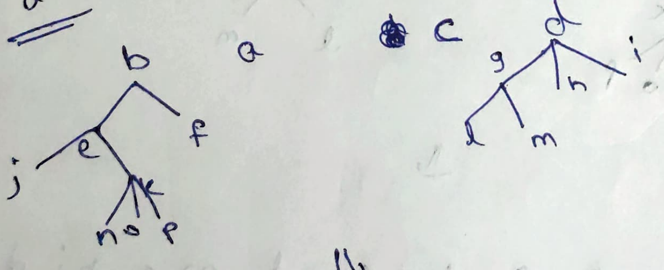
* Visit left most subtree, visit root, visit other subtree left to right.

1st



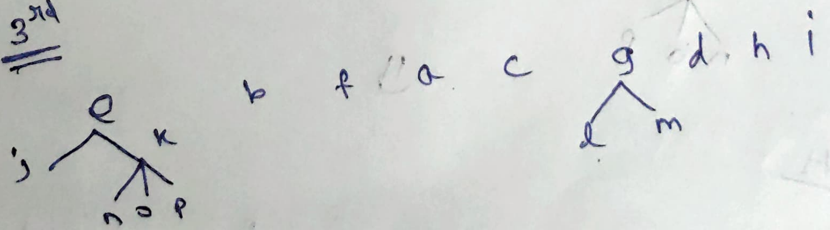
⇓

2nd



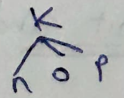
⇓

3rd



⇓

4th

j e 

b f a c l g m d h i

⇓

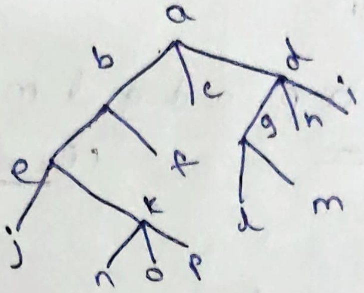
5th j e n k o p b f a c l g m d h i

(In order)

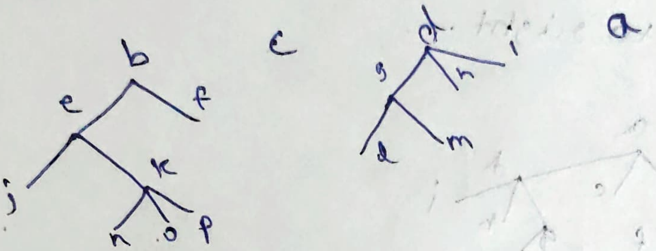
For Post Order Traversal

* Visit Subtree left to right, visit root

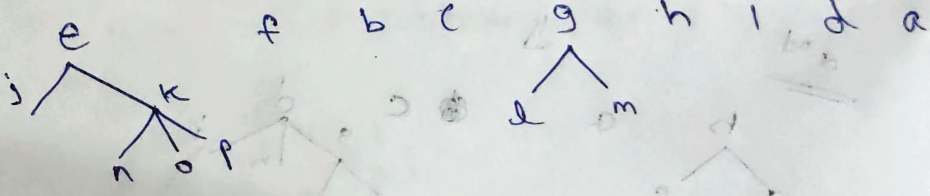
1st



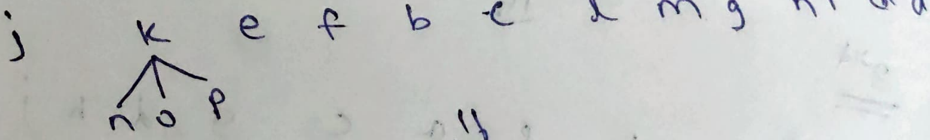
2nd



3rd



4th



5th

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

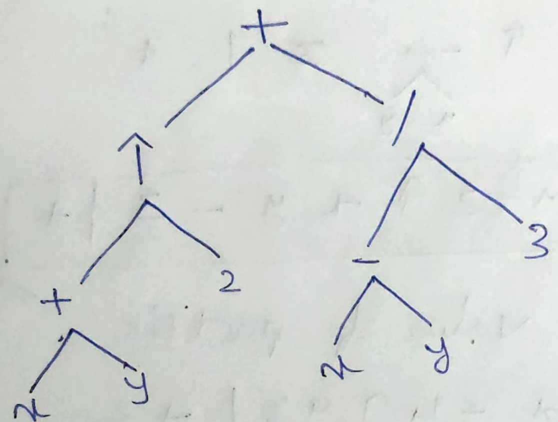
(Post order)

Infix, Prefix and Postfix notation

* An ordered rooted tree can be used to represent such expressions, where the internal vertices represent operations and the leaves represents the variable or numbers. Each operation operates on its left & right subtree.

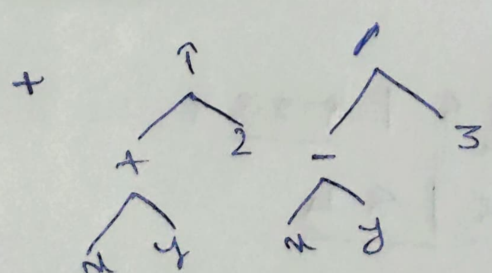
* Ex
  What is ordered rooted tree that represents the expressions

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



* Prefix form can be obtained after applying Preorder traversal.
* Infix form can be obtained after applying In order traversal.
* Postfix can be obtained after applying post-order traversal.

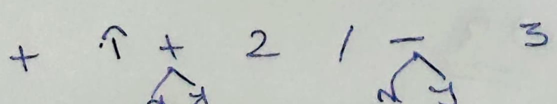
2nd



(Prefix).

⇓

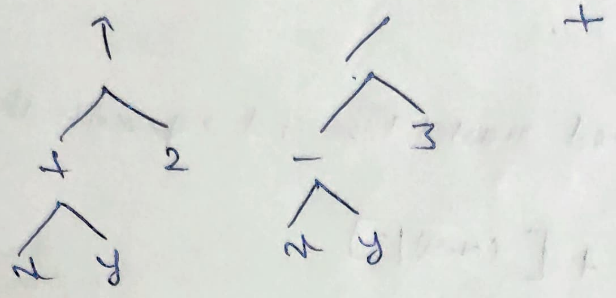
3rd



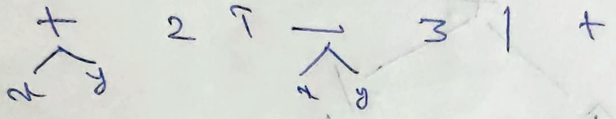
$\Downarrow$

$+ \uparrow + x\, y\, 2 \,/\, - x\, y\, 3$

Post fix



$\Downarrow$



$\boxed{x\, y + 2 \uparrow x\, y - 3 \,/\, +}$

* What is the value of post fix

$7\ \underbrace{2\ 3\ *}\ -\ 4 \uparrow 9\ 3\ /\ +\ 2$

$7\ 6\ -\ 4 \uparrow 3\ +$

$1\ 4 \uparrow 3\ +$

$1\ 3\ +$

$\Rightarrow 4$

Prefix expression

$+\ -\ \underline{*\ 2\ 3}\ 5\ /\ \underline{\uparrow 2\ 3}\ 4$

$+\ \underline{-\ 6\ 5}\ \underline{/\ 8\ 4}$

$+\ 1\ 2$

$\Rightarrow \underline{3}$