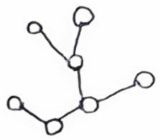
Trees:-

Def. A Tree T is acyclic (has no cycles) Connected graph.

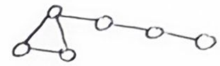
Example:-

T_1: 

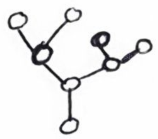
is a tree.

T_2: 

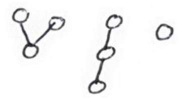
is a tree
"trivial tree"

T_3: 

is not a tree
Since it has a cycle.

T_4: 

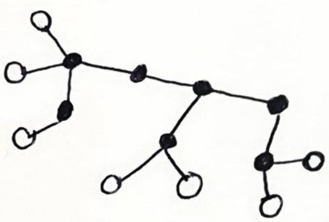
is a tree

T_5: 

is not a tree.
Since it's not connected.

Note that:-
1. Size of T = order of $T-1$ i.e $|E(T)| = |V(T)| - 1$
2. If T is a non-trivial tree, then it has at least two end-vertices "leaves"
3. If T is a tree, then $\forall x, y \in V(T)$, $\exists!$ x-y path in T.
4. Every non-trivial tree must have (i) a leaf "vertex with degree 1"
   (ii) internal vertecies ((vertices with degree ≥ 2))

Example:-



This is a tree with 7 leaves
and 7 internal vertices.

so $|E| = 13$ and $|V| = 14$

Example: Let T_1 be a tree with size 13 and Let T_2 be a tree with order = 3 times of order of T_1. Find the order of T_2 and it's size.

$|V_{(T_1)}| = 13 + 1 = 14$

$|V_{(T_2)}| = 3|V_{(T_1)}| = 3(14) = 42$

so $|E_{(T_2)}| = |V_{(T_2)}| - 1 = 42 - 1 = 41$

There are two types of Trees we are intrested in:-
(1) Rooted trees. (2) Spanning trees.

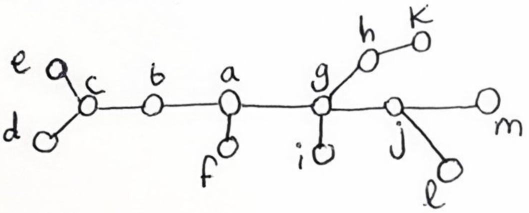
Rooted Trees :-

Def. A tree in which one vertex has been designated as a root (start point) and every edge is directed away from the root.
We typically place the root at the top of the tree.
denoted by (T, v_0). where v_0 is the root.

Note that :- Any vertex can be the root of the tree whether it's a leaf or an internal vertex.
So in a tree T of order n, we have n different rooted trees.

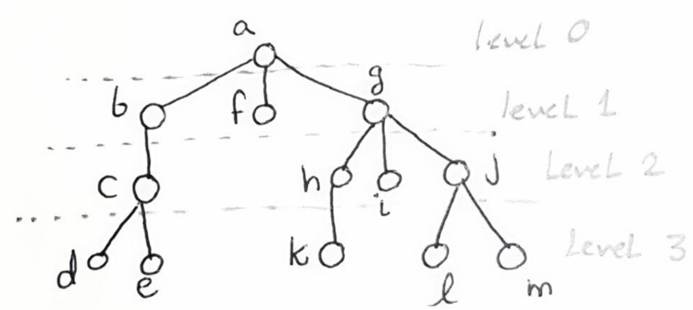
Examples :- Let T be the following tree.

T:



T is of order 13 and size 12.

If we design $\underline{\underline{a}}$ as a root (T, a), then the rooted tree will be as follows.

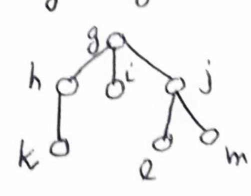


* According to the distance betwen the vertex and the root we can divide the tree into levels.

Note :- The Largest level is called the hight of the rooted tree.
i.e the max $(d(v, v_0))$ where v is a leaf and v_0 is the root.

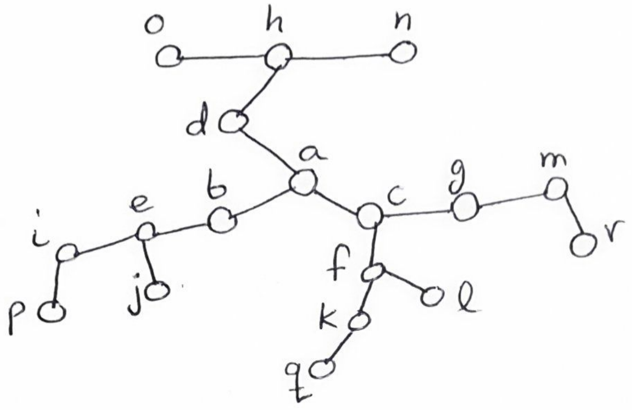
Important Consepts :-

1. parent : The first vertex up to the root. — in the above example. parent of c is b
2. offspring ((child)) : The first vertices down to the leafs. — offspring of g are h, i & j
3. siblings :- vertices of the same parents. — siblings of h are i & j
4. Descendents :- all vertices down to the leafs. — desendents of b are c, d & e
5. Subtree : the vertex and it's desendents. — subtree of g $T(g)$



Example:- Let T be the following graph.

T:



Let a be the root of this tree.

Find:-

Parent of f, the offspring of f, the siblings of f, the descendents of f and the subtree of f.

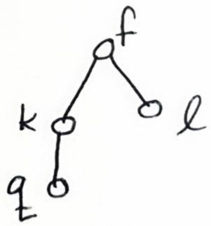
parents of f is C

The off spring of f are k and l.

The siblings of f is g

The desendents of f are k, q and l.

The subtree of f T(f)



Types of rooted trees:-

[1] n-ary tree.

A rooted tree in which every vertex has no more than n offsprings.

[2] Complete n-ary tree.

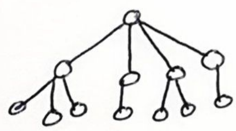
A rooted tree in which every internal vertex has exactly n offsprings.

Note that : [1] If $n=2$ we call [1] a binary tree and [2] complete binary tree.

[2] It used in AI and in describing sequences of yes/no desicions.

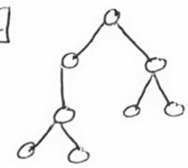
Example:

[1]



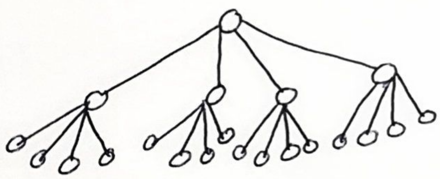
4-ary tree

[2]



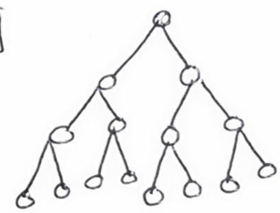
2-ary. tree
"binary tree"

[3]



Complete 4-ary tree.

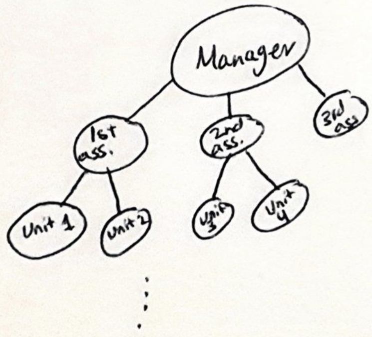
[4]



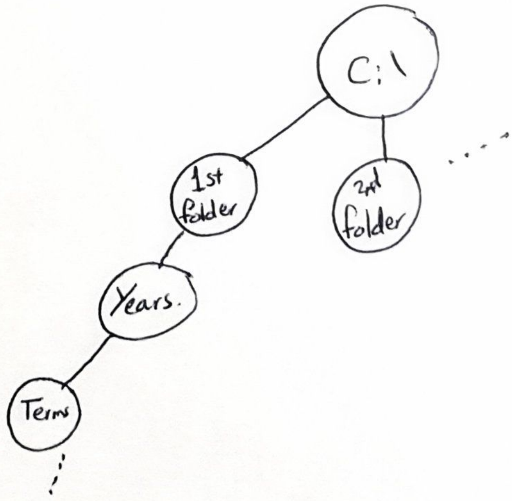
Complete 2-ary tree
"Complete binary tree"

• We use rooted trees to model:-

(1) Organizational structures.



(2) Computer files systems.



Positional (ordered) Rooted tree:-

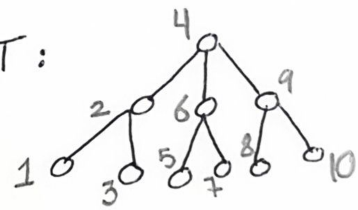
It used in expeint systems. when there are more important questions.

It's Just a rooted tree in which the offspring of each internal vertices are ordered.

* To order a binary tree define left offspring — right offspring.
  and left subtree — right subtree.

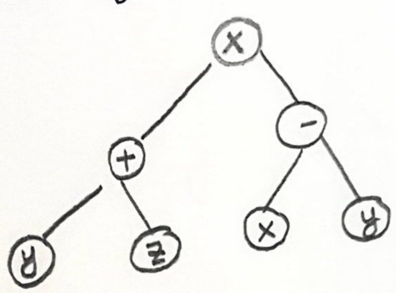
* To order an n-ary tree with $n > 2$ left most — right most.

Example:- Order T :

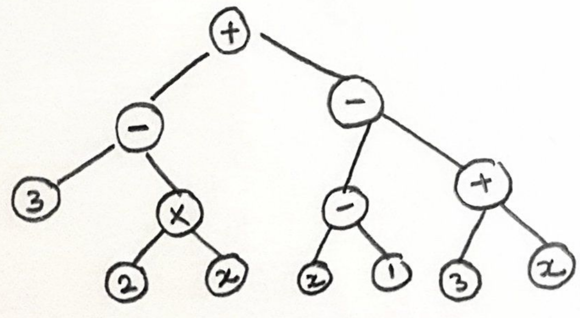


Example:- Find the arithmetic expression tree of:-

1. $(y+z) \times (x-y)$



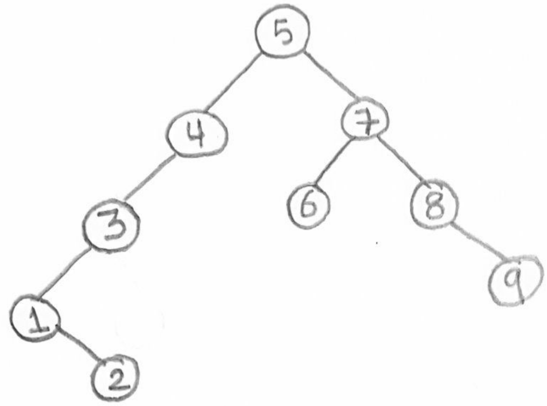
2. $(3-(2 \times x)) + ((x-1)-(3+x))$



Form a binary search tree for:-

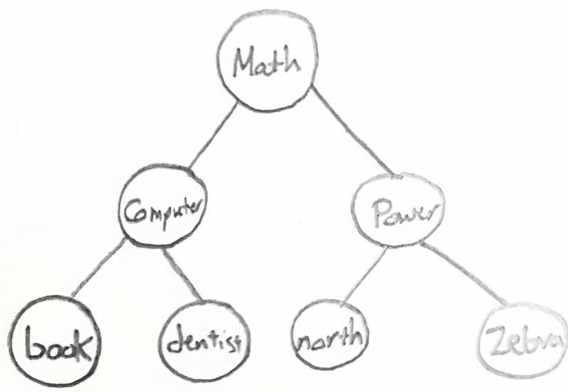
[1] 5, 7, 4, 3, 6, 1, 2, 8, 9

Start



[2] Math, Computer, power, north, Zebra, dentist, book.

Start



Tree Searching :-

A process of visiting the vertices in a tree once and only once.

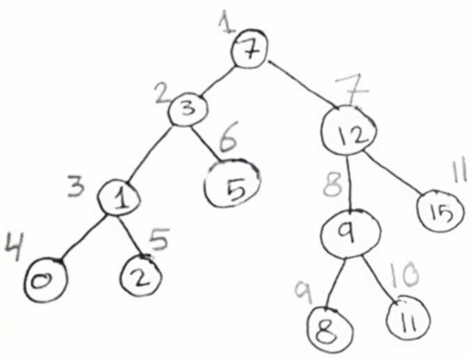
There are 3 types of tree searches:-

[1] Preorder Search [2] Inorder Search [3] post order search

[1] Preorder Search:-

follow this algorithm:- visit the root go to left subtree then the right subtree.

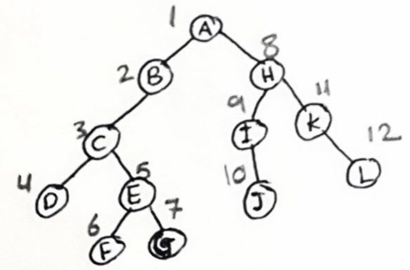
Example : [1]



So the result

7, 3, 1, 0, 2, 5, 12, 9, 8, 11, 15

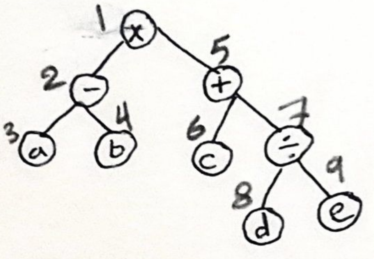
[2]



So the result

A B C D E F G H I J K L

[3]



so the result

X − a b + c ÷ d e "prefix or Polish form"

If a=6, b=4, c=5, d=2 & e=2 then X − 6 4 + 5 ÷ 2 2

X 2 + 5 ÷ 2 2

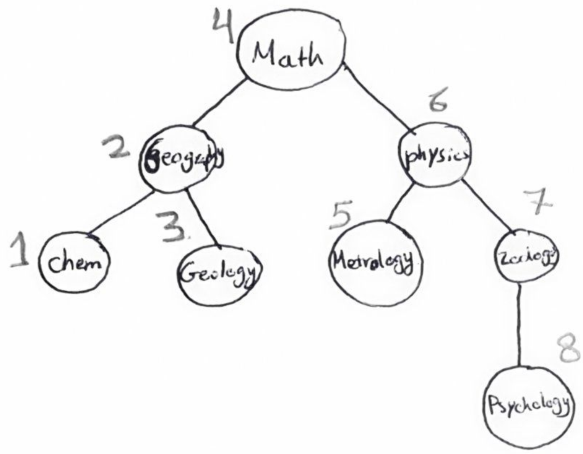
X 2 + 5 1

X 2 6 = 12

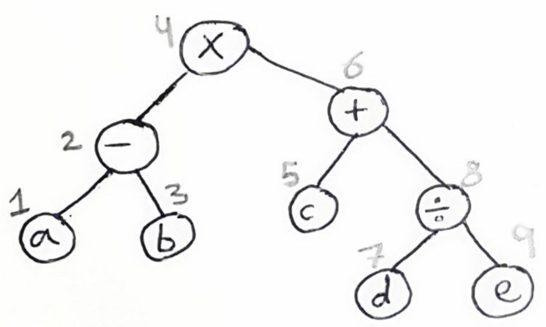
[2] Inorder Search:-

The algorithm is: search the left subtree, visit the root then search the right subtree.

Example: [1]



[2]



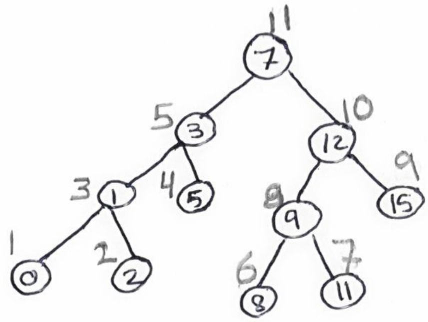
a − b × c + d ÷ e "Infix"

we can't evaluate the arithmatic operation since we can't determine the exact expression.

[3] Postorder search:-

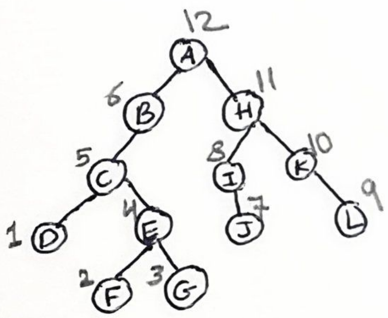
The Algorithm is:-
search the left subtree, search the right subtree & then visit the root

examples [1]



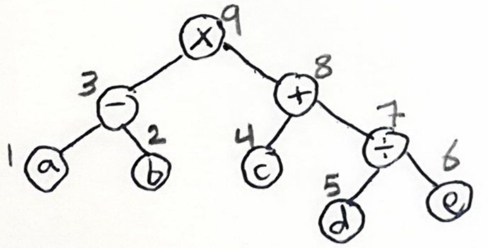
So, the result.
0, 2, 1, 5, 3, 8, 11, 9, 15, 12, 7

[2]



The result:-
D F G E C B J I L K H A

[3]



a b - c d e ÷ + × "postfix or Reverse Polish Form"

If a = 6, b = 4, c = 5, d = 2 & e = 2

6 4 - 5 2 2 ÷ + ×

6 4 - 5 1 + ×

6 4 - 6 ×

2 6 ×

12