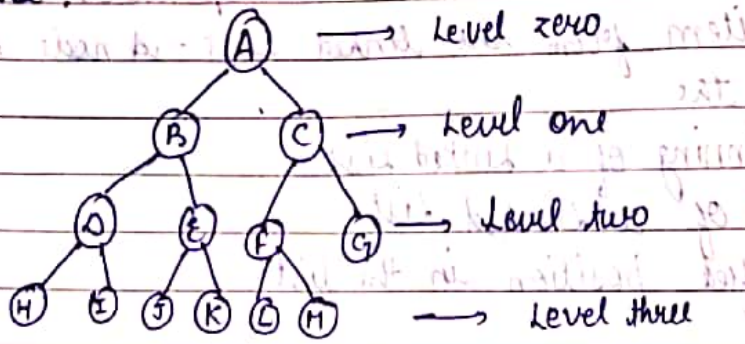
UNIT 3: TREES

A Tree is a non-linear data structure in which items are arranged in a sorting sequence. It is used to represent hierarchical relationship existing among several data items.

- Tree is a finite set of one or more data items such that where is the special data item called the root of the tree and its remaining data are partitioned into number of disjoint subset each of which is itself a tree which is known as subtree.



Tree Terminology :-

1. Roots :- It is specially designed data item in a tree. It is a first item in their hierarchical arrangement of data item.
In the this fig: (A) is the root node

2. Node :- Each data item in a tree is called a node. it specified the data information or a link to the other data item.
There are 13 node in the above tree

3. Degree of Node :- It is number of subtree of a node.
In above tree the degree of node are

A = 2 G = 0 M = 0
B = 2 H = 0
C = 2 I = 0
D = 2 J = 0
E = 2 K = 0
F = 2 L = 0

Degree of Tree:- It is the main degree of node in the above tree maximum degree of a node is 2 so the

Terminal Node:- A Node with degree zero is called a terminal node or leaf node.
G, H, I, J, K, L, M are terminal node of above tree.

Non-terminal Node:- Any Node Exepect the root node whose degree is not zero is called non-terminal node.
(B, C, D, E, F).

Siblings:- The children node of a given Parent node is called siblings they are also called Brothers.
BC is a sibling of A
BE is a sibling of B
FG is a sibling of C
HI is a sibling of D
JK is a sibling of E.

Level
The entire tree structure is labelled in sucha way that the root node is always zero. then its Immediate Children are at level 1 & their Immediate children are at level 2 & so on. upto the terminal node.

Edge :- Edge is a connecting line of 2 nodes that is line drawn from one node to another node is called an Edge.

Path :- It is a sequence of consecutive edge from the source node to the destination node in the above field. The Path b/w A & M is given by the node pair.
(AC),(CF),(FM)

Height / Depth :- It is the main level of any node in given tree.

Edge / Height = Level + 1
Height = 3 + 1
Height = 4

Forest (Disjoint subsets) :-
It is a set of disjoint tree in a given tree if you remove its root node then it become a forest.

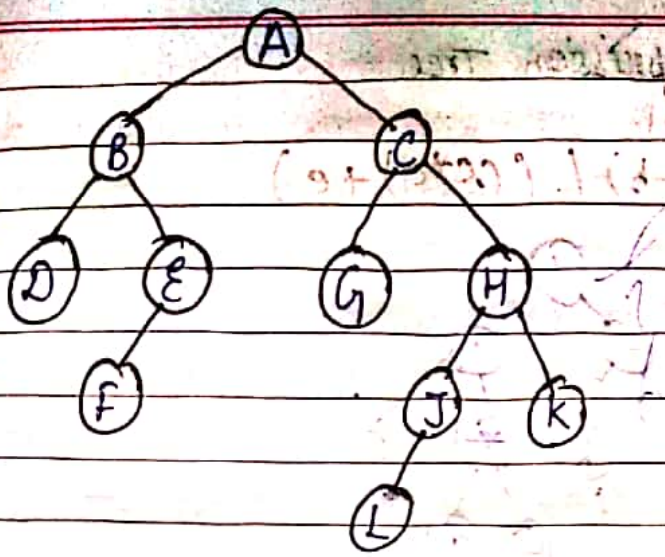
Imp **Binary Tree :-** A Binary Tree is defined as finite set of elements called nodes such that :

(a) T is Empty (called the null tree or the empty tree)
OR
(b) T Contains a distinguish nodes R, called the Root of T & the remaining nodes of T from an ordered pair of disjoint binary trees T_1 & T_2

If T does contain a root R then the two trees T_1 & T_2 are called respectively the left & right subtrees of R.

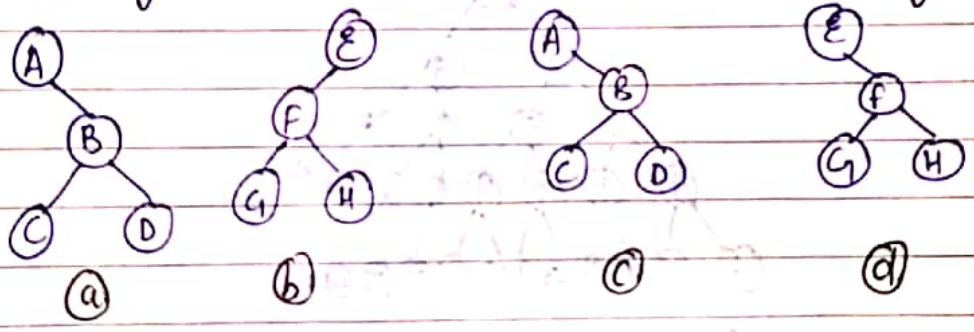
- ~~If T_1 is non-empty~~
- Any Node N in a Binary Tree has either 0, 1 or 2 successors

Imp



T Consist of 11 nodes B is left successor & C is a right successor of the node A
The left subtree of the root A Consist of the node B, D, E, and F And the right subtree of the root A Consist the node C, G, H, J, K & L.

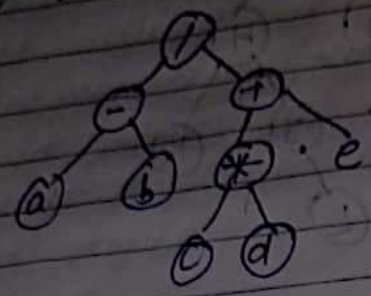
Binary Tree T & T' are said to be similar if they have the same structure Or in other words they have the same shape. The Trees are said to be Copies if they are similar and if they have the same Contents at Corresponding nodes.



The Three trees a, c, & d are similar because the tree a and c are Copies since they also have the same data at Corresponding nodes.

Algebric Expression Tree

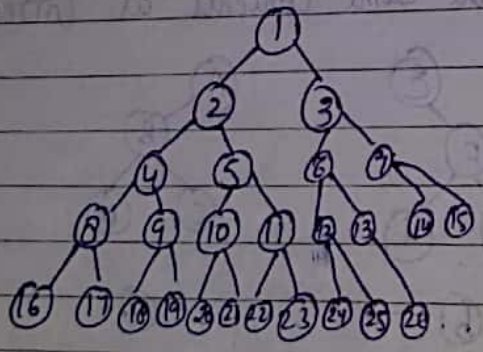
Ex→ $E = (a-b) / ((c*d)+e)$



Complete Binary Tree : (0-2)

Consider any Binary Tree T each node of T can have atmost two children.

- The Tree T is said to be Complete if all its levels Expect possibly the last have the maximum number of possible nodes and if all the nodes appear as far left as possible thus there is unique complete tree T_n with exactly n nodes



Complete Tree T_{26}

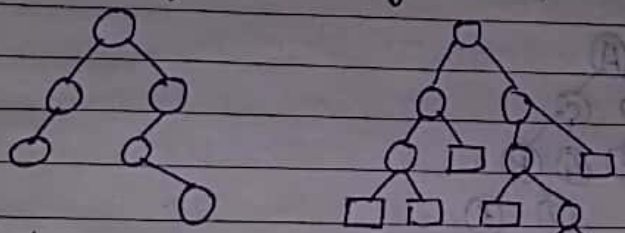
Extended Binary Tree (2-tree)

A Binary Tree is said to be a 2-tree or an extended Binary tree if each node (N) has either 0 or 2 children.

In such a case the nodes with 2 children are called Internal nodes and the nodes with 0 children are called External node.

Sometimes the nodes are distinguish in diagram by using circles for Internal node and squares for External nodes.

eg→ converting a Binary Tree into a 2-tree :-



(a) Binary Tree

(b) Extended Binary Tree

Representing Binary Tree in Memory :-

Let 'T' be a Binary Tree which can store in a Memory in two ways.

(a) The First is link representation of T.

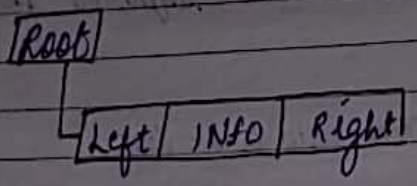
(b) The second way which use a single array called the sequential representation of 'T'

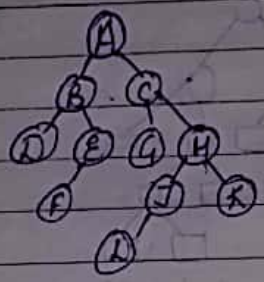
The Main requirement of any representation of 'T' is that one should have direct access to the root R of T and given any node of T one should have direct access to the children of T.

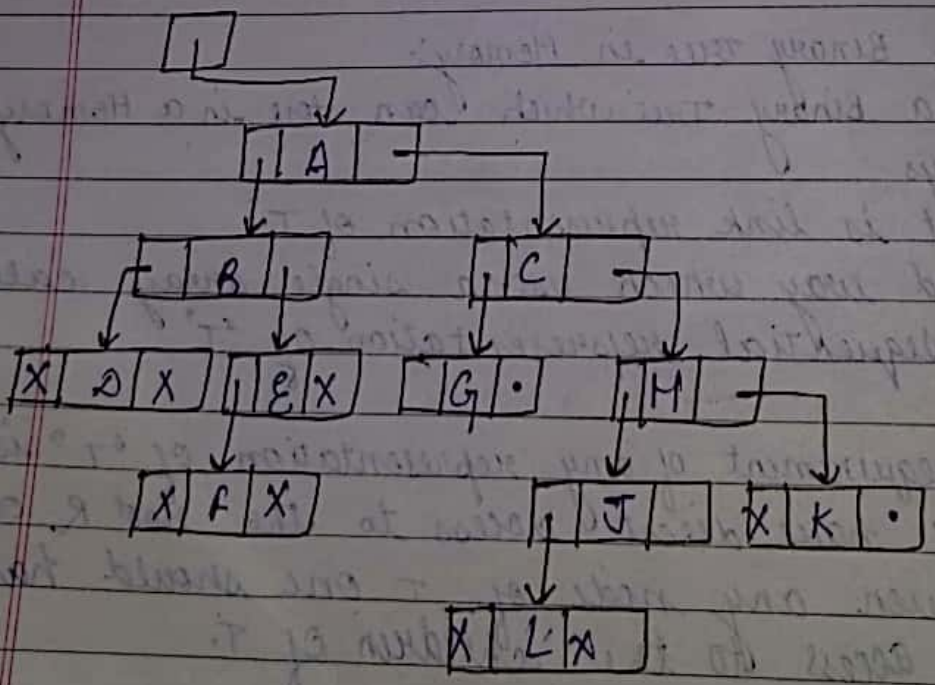
Link Representation of Binary Tree

Link Representation of Binary Tree use 3 parallel arrays INFO, LEFT And RIGHT and a pointer variable ROOT as follows Each node N of T will Correspond to a location K such that

(a) INFO[K] → Contains the data at the Node N.
(b) LEFT[K] → Contains the location of left children of Node N.
(c) RIGHT[K] → Contains the location of the Right child of Node N.

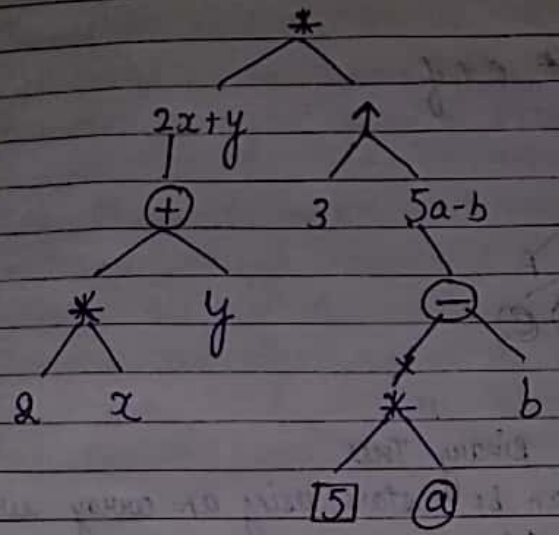




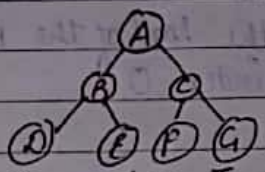


Consider the algebric Expression

$E = (2x+y)(5a-b)^3$



→ Consider the following Binary Tree T.



For the above tree T find the values for the following

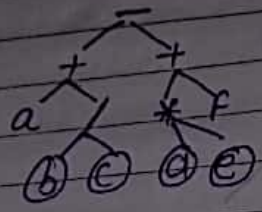
① root node A
② leaf node D, E, F, G
③ Pair of Siblings DE is the sibling of B
FG is the sibling of C
BC is the sibling of A

④ Depth of Tree 3
⑤ Parent node of G → C
⑥ child of node A BC

(i) Any path in the tree →
AC, CF

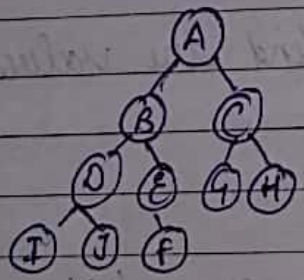
(ii) level of the tree → 2.

Ques-1 $\varepsilon = \varepsilon a + b / c - d * e + f$



Array Representation of Binary Tree

A Binary tree can be store using an array which is called a sequential representation of Binary Tree. This type of representation is static i.e a block of memory for an array is to be allocated before going to store the actual tree into it. An array can be used to store the load of the Binary tree the root node is always at index 0.



A	B	C	D	E	G	H	I	J	F