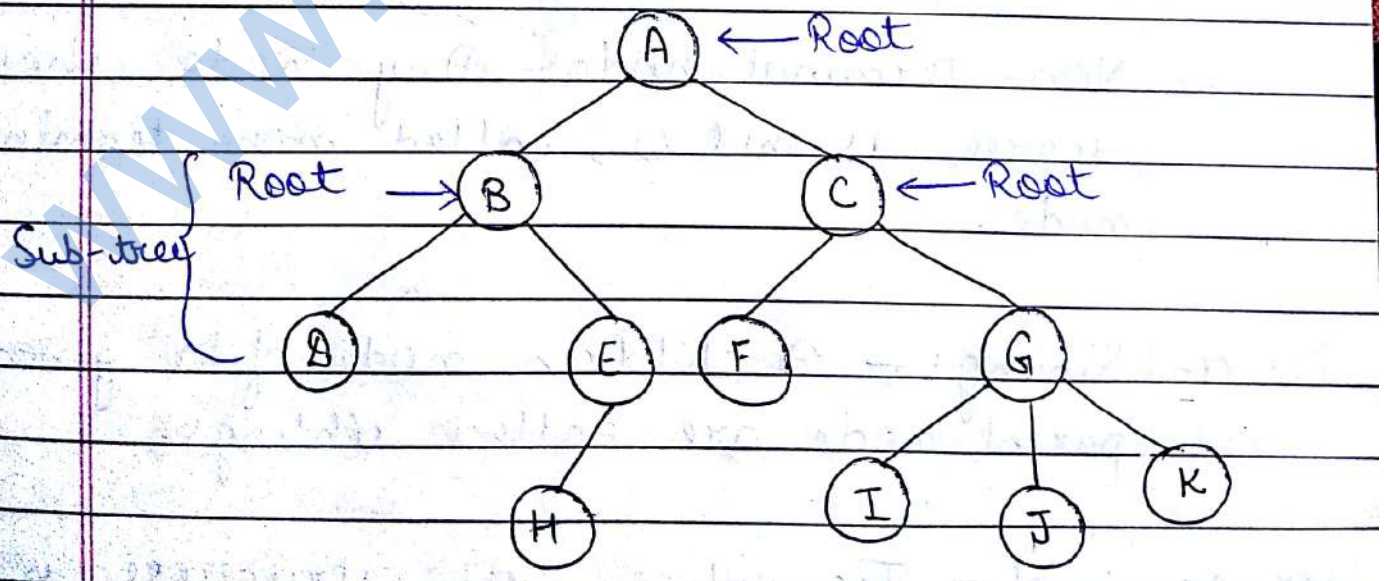
UNIT - 4

Tree

- A tree is a non-linear data-structure in which items are arranged in a sorted sequence.
- It is used to represent hierarchal relationship existing among several data-structures.

Tree:- It is a finite set of one or more data-items, such that -
There is a special data-item called the root of the tree.
And its remaining data-items are partitioned into number of mutually exclusive subsets, each of which is itself a tree, and they are called subtree.



Date / /

★ Tree terminology :-

1- Root – It is specially designed data-item in a tree. It is at the first in the hierarchical arrangement of data-items.

2- Node – Each data-item in the tree is called a node. It is the basic structure in the tree. It specifies the data information and link to other data items.

3- Degree of a node – It is the number of sub-tree of a node in a given tree.

4- Degree of a tree – It is the maximum degree of node in the given tree.

5- Terminal node – A node with degree 0 is called terminal node.

6- Non-terminal node – Any node whose degree is not 0, called non-terminal node.

7- Siblings – A children node of a given parent node are called siblings.

8- Level – The entire tree structure is leveled in such a way that the root node is always at level 0.

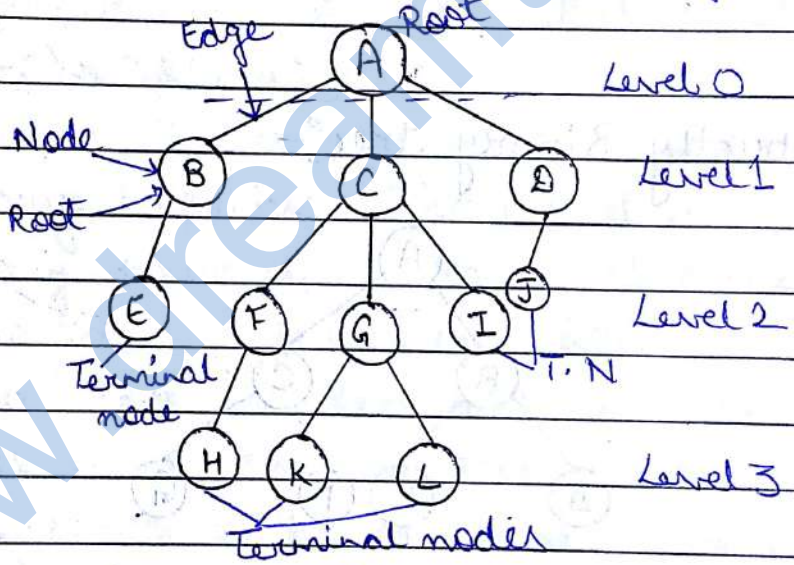
Date / /

9- Edges – It is a connecting line of two nodes.

10- Path – It is a sequence of consequtive edges from the source node to the destination node.

11- Depth – It is the maximum level of any node in a given tree. It is also known as height of a tree.

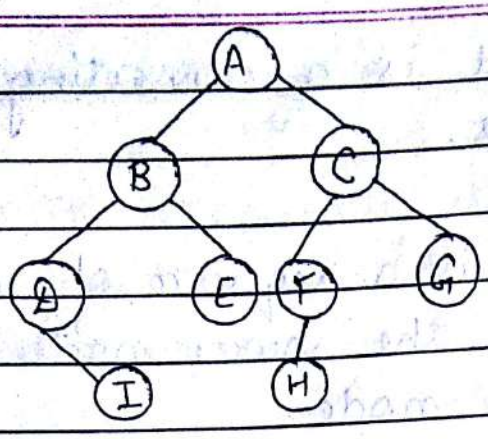
12- Forest – It is a set of disjoint tree.



Degree of tree = 3

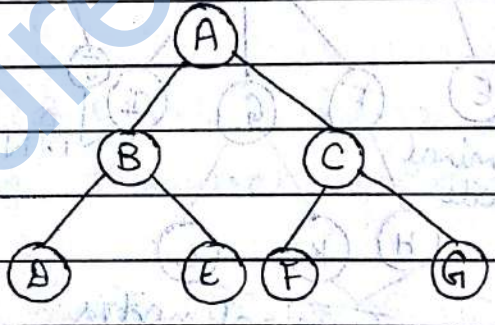
* Binary tree :- A binary tree is a set of finite data items which are either empty or consist of a single item called the root and two ~~sets~~ disjoint binary tree called the left sub tree and right sub tree.

Date / /



1. A binary tree is a very important and the most commonly used non-linear data-structure.
2. In a binary tree, the maximum degree of any node is atmost 2.
3.

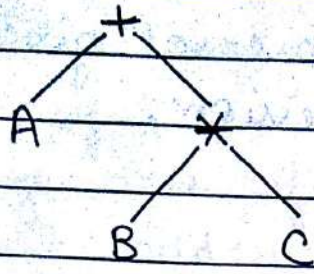
* Strictly Binary tree:-



- If every non-terminal node in a binary tree consist of non-empty left sub-tree and right sub-tree, then such a tree is called strictly binary tree.
- In the above binary tree all the non-terminal nodes having non-empty left or right sub-tree.
- An expression can be represented with the help of binary tree.

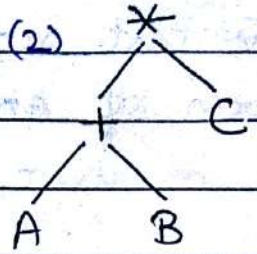
Date / /

(1)



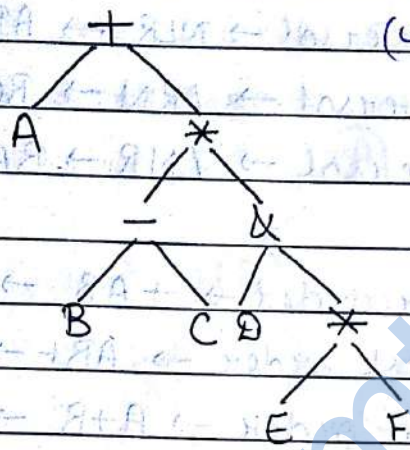
A + (B * C)

(2)



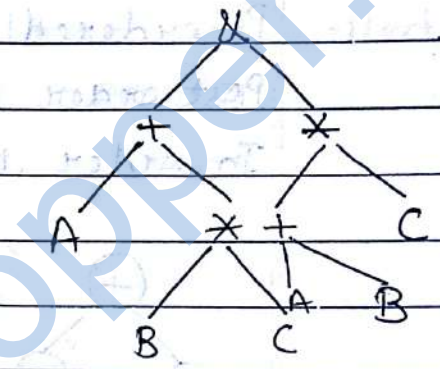
(A+B) * C

(3)



A + (B - C) * D & (E * F)

(4)



(A + (B * C)) & ((A + B) * C)

★ Binary Expression Traversal :-

Binary expression traversal gives the prefix and infix expression.

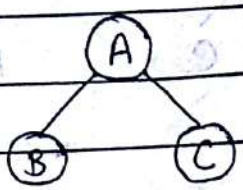
When tree is traverse in pre-order means that the operator precides to operate. Thus the pre-order traversal gives the prefix for the expression.

1. + A * BC
2. * + ABC
3. + A * - BC $ D * EF
4. $ + A * BC * + ABC

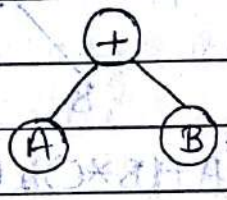
Traversing a binary expression in post order places an operator in postfix form.

Date / /

The post order traversal gives the postfix form of an expression.



Preordered traversal → NLR → ABC
Post order traversal → LRN → BCA
In order traversal → LNR → BAC

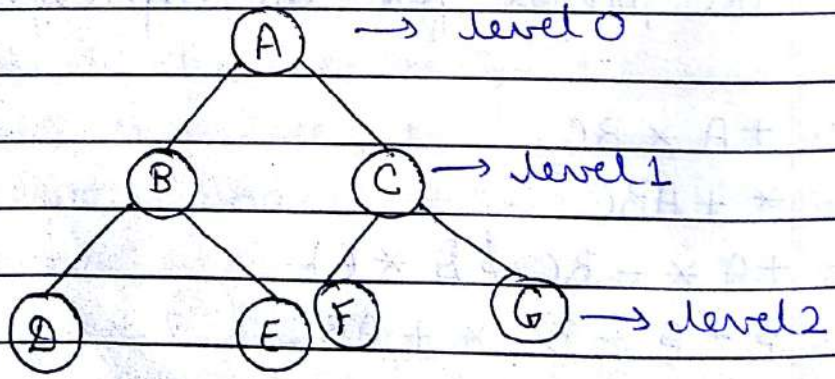


Preorder → +AB → NLR
Post order → AB+ → LRN
In order → A+B → LNR

* Complete binary tree :-

1- A binary tree with n nodes and of depth d is a strictly binary tree, all of whose terminal nodes are at level d.

2- In a complex binary tree, there is exactly 1 node at level 0, 2 nodes at level 1 and 4 nodes at level 2.

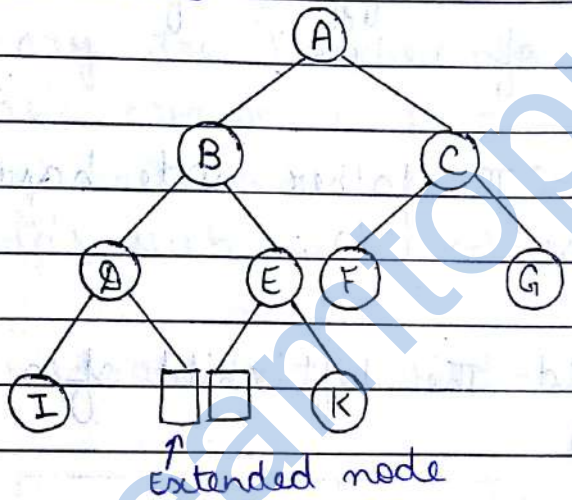


Complete binary tree

Date / /

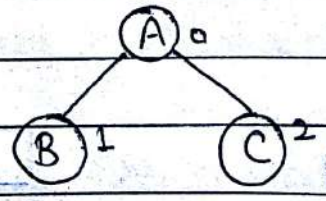
* Extended binary tree -

- A binary tree is said to be extended binary tree, if each node n has either 0 or 2 children.
- In such a case the nodes with two children are called internal nodes and the nodes with 0 children are called external nodes.



* Binary tree representation :-

1- Array representation - An array can be used to store the nodes of a binary tree. The nodes stored in an array are accessible sequentially.

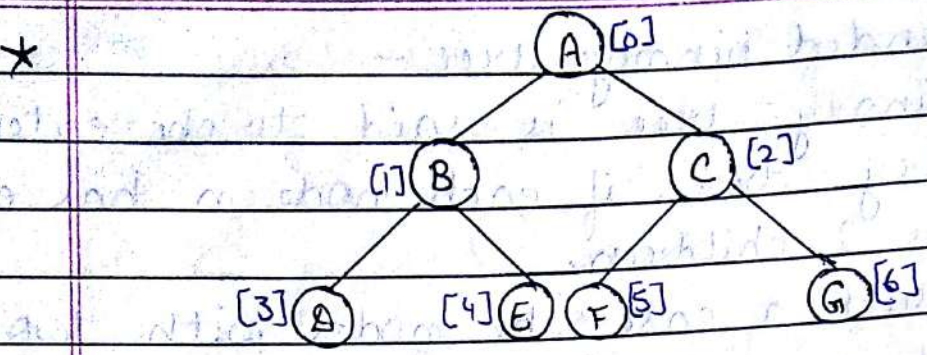


Binary tree

a[0]	A
a[1]	B
a[2]	C

Date / BT

*



	BT
0	A
1	B
2	C
3	D
4	E
5	F
6	G

How to identify the father, children and siblings of a tree?

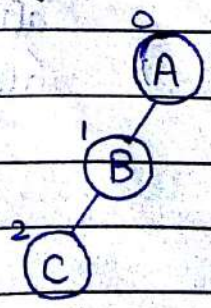
1. Father – The father node have an index is at floor $(n-1)/2$.

2. Left child – The left child of a node n is at $(2n+1)$.

3. Right child – The right child of a node is $(2n+2)$.

4. Sibling – If the left child at index n is given, then its right sibling is an $(n+1)$. Similarly, if right child is given at index n, then its left sibling is at $(n-1)$.

- Disadvantage –



A
B
C

Date / /

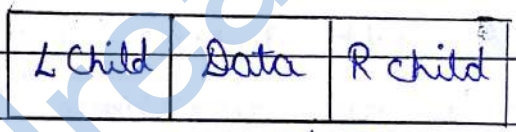
The array representation is more ideal for complete binary tree, but it is not suitable for other than complete binary tree as it results in unnecessary wastage of memory space.

2. Link-list representation of binary tree -

→ To avoid memory loss link-list is used for binary tree representation.

→ The basic component to be representate in a binary tree is a node. The node consist of 3 fields such as -

- Data
- Left child and right child



(Node structure Of Binary Tree)

Logical representation -

```
struct node
{
char data;
struct node *l child;
struct node *r child;
};
typef struct node node;
```

Date / /