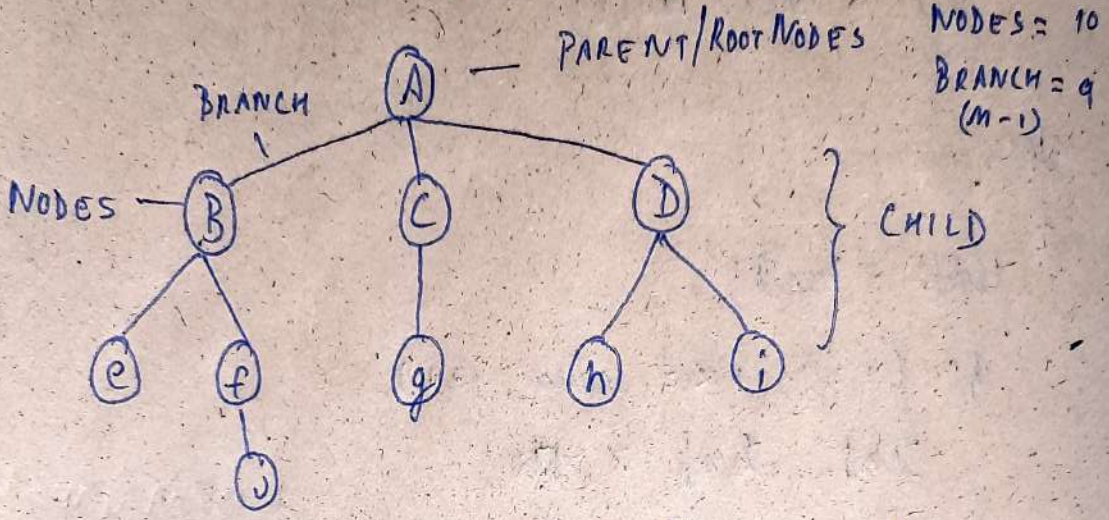
06-08-2024

NON-LINEAR DATA STRUCTURES =>

* Trees * Graphs

1. TREES :- It shows parent - child relationship.



BRANCH :- It is the connection between 2 nodes.
If a tree contains 'n' nodes then it contains (n-1) branches.

LEAF NODE :- A node doesn't have any children.

Ex :- 'e' node

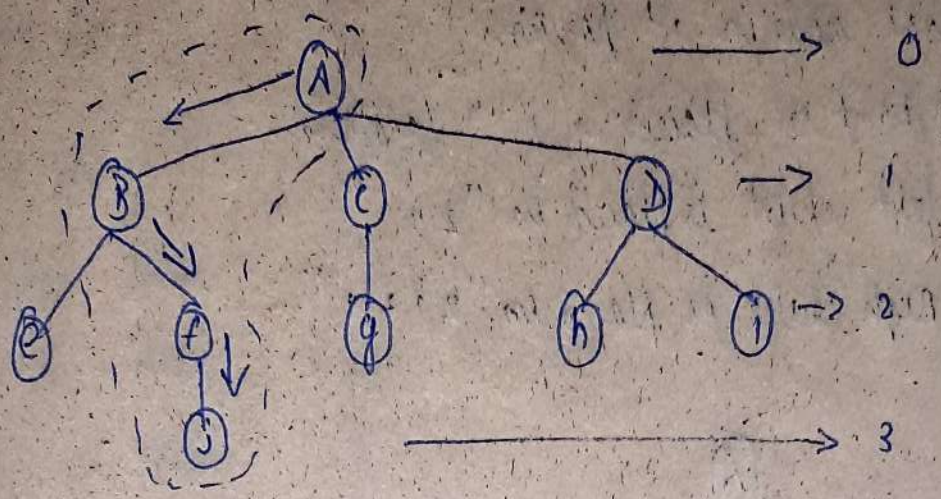
The degree of a leaf node is always zero [0]

DEGREE OF A NODE :- The no. of children to a particular node is degree of that node.

The degree

ORDER OF A TREE :- It is the highest degree of particular node

HEIGHT / DEPTH OF A TREE :-

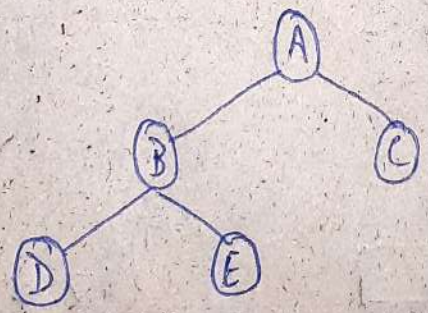


It is the longest path from Root node to any leaf node.

BINARY TREE :- the order of degree = 2

A binary tree is either empty or consists. If consists it contains a special node called Root node, remaining nodes are partitioned into left & right sub-trees.

Sub-trees themselves follow the rules of Binary Tree



* Arrays
* Linked List

Representation of Binary Tree :-

-> Arrays

-> Linked List

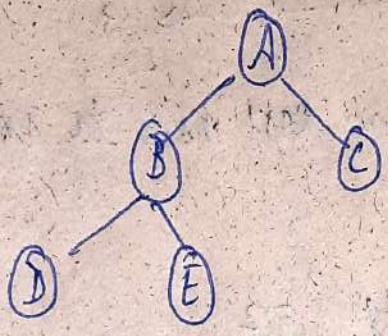
Array Representation :-

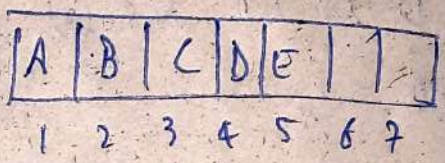
→ Place root node in 1st position

→ If a patent is placed in i th location

its left child is placed in "2i"

Righ child is placed in "2i+1"





Advantages :-

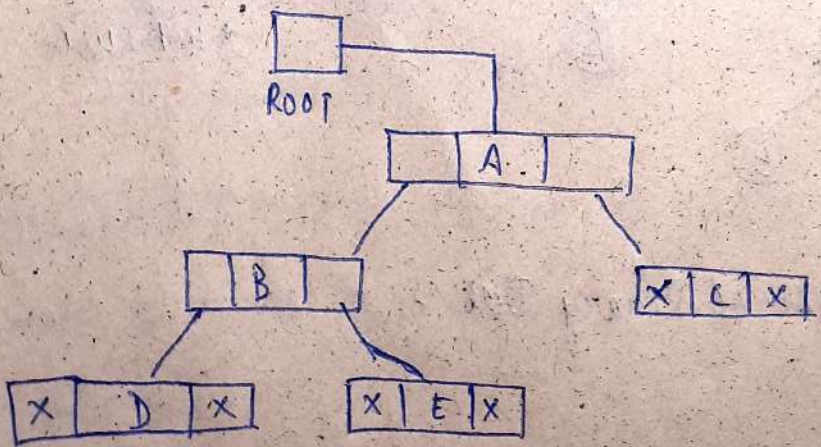
* Finding Parent child is easy

4 → Parent/Root node

$x2$ = Parent

$\div 2$ = child

Linked List Representation :-



X → NULL

OPERATIONS ON BINARY TREE =>

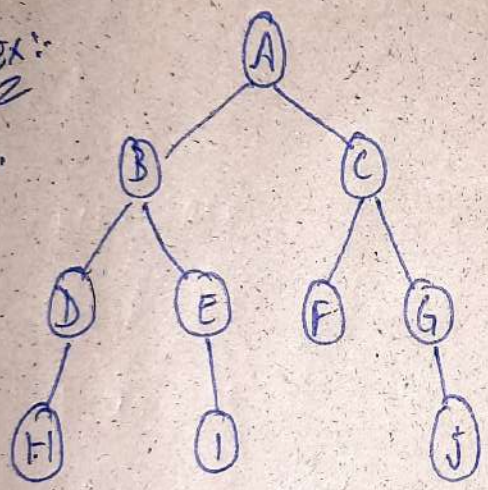
* Insertion * Deletion * Traverse

TRAVERSAL ORDERS OF BINARY TREE =>

1. Pre - order = + A B.
2. In - order = A + B → *
3. Post - order = A B +

EX:-

1.



1. PRE-ORDER =

A B D H E I C F G J

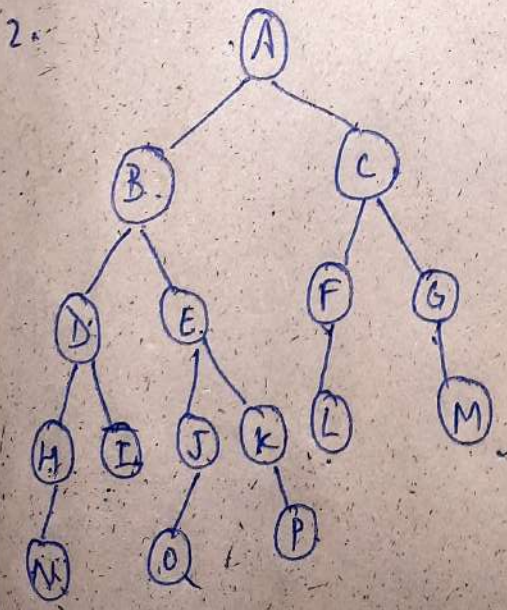
2. IN ORDER =

H D B E I A F C G J

3. POST ORDER =

H D I E B F J G C A

2.



1. PRE - ORDER =

A B D H N I E J O K P C F L G M

2. IN - ORDER =

~~N H D I B O J O P K L F C M G A~~

N H D I B O J E K P A L F C G M

3. POST ORDER =

N H I D O J P K E B L F M G C A