Non-linear Data structures

Page No.
Date

* Tree

Node: This is the main component of tree structure. A node of tree stores actual data and links to the other node.

Parent Node: Parent of node is the immediate tree processor of node

child node: Immediate successor of a node are known as child.

Link: This is a pointer to a node in a tree. LC & RC are two link of a node.

root: This is a specially designated node which has no parent.

Leaf: The node which is at the end and doesn't have any child - [terminal node] / [External node]

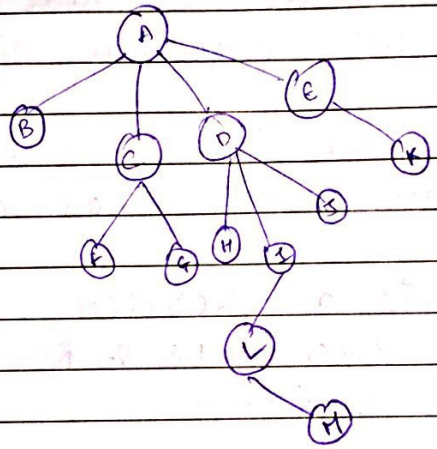
Level: It is the rank of the hierarchy. The root node has level 0.

Height: Maximum number of node that is possible in a path starting from root node to leaf node.

Page No.
Date

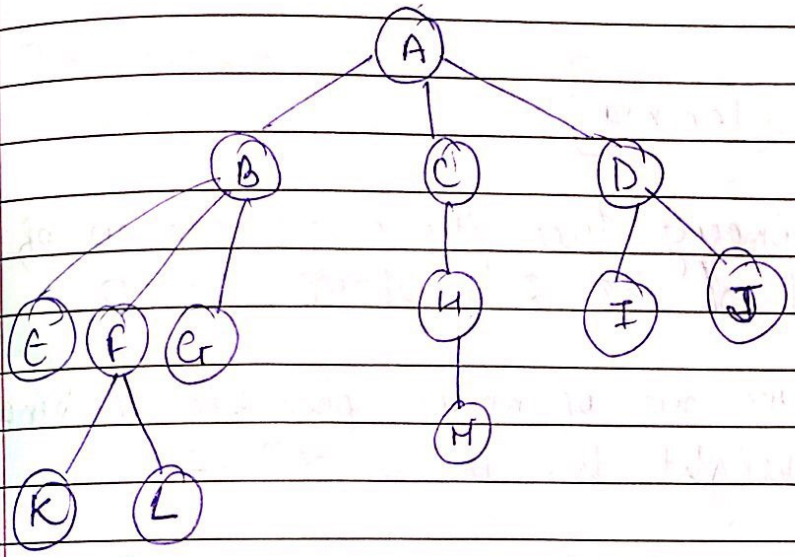
Degree : The maximum no. of children that is possible for a node.

Sibling : The nodes which have same parent.



⇒ Observe the Given tree and find the following with the reference tree.

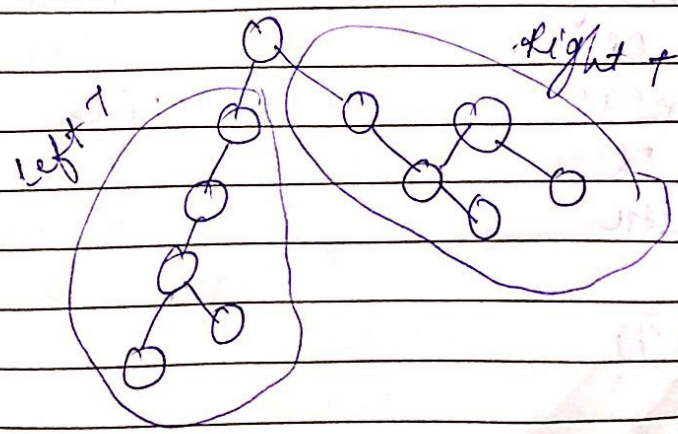
1. height → 4 ✓
2. level of H → 2 ✓
3. level of C → 2 ✓
4. level of K → 3 ✓
5. Degree → 3 ✓
6. longest path in a tree → ABFK, ABFL, DCHM
7. Parent (M) → H
8. Sibling (i) → J
9. child (b) → E, F, G



Binary Tree : A binary tree is a special form of a tree. It can be define as finite set of node such that

(i) Tree is Empty

(ii) T contains a specially designated node called root of T and the remaining node of T form two disjoint binary tree T_1 and T_2 which are called of ~~legs~~ of Left Sub Tree & Right Sub tree.



Page No.
Date

Properties : [Lemma]

1) In any binary Tree, the maximum no. of nodes on level k is 2^k, where $k \geq 0$

2) The maximum no. of nodes possible in binary tree of height h is $2^h - 1$

3) For any non-empty binary tree T if with n_0 is the number of leaf nodes with degree 0 and n_2 is the number of internal node, degree 2. then $n = 0$ is $n_2 + 1$

$$n_0 = n_2 + 1$$

=> Representation

- Sequential
  - OR
  - Array
  - OR
  - Linear.
  - OR
  - Static
- Link
  - OR
  - Dynamic
  - OR
  - Link list

The minimum number of nodes in a binary tree of height is h

For any non-empty binary is n is the no. of nodes and e is the no. of edges then $n = e+1$

Following Rules can be use to decide location of any node of a tree in the array (Assuming that array index starts from 1)

1: The root node is at location one
2: For any node with index i, $1 < i \leq n$:

2A :- Parent (i) = $i/2$

2B :- LChild (i) = $2 * i$

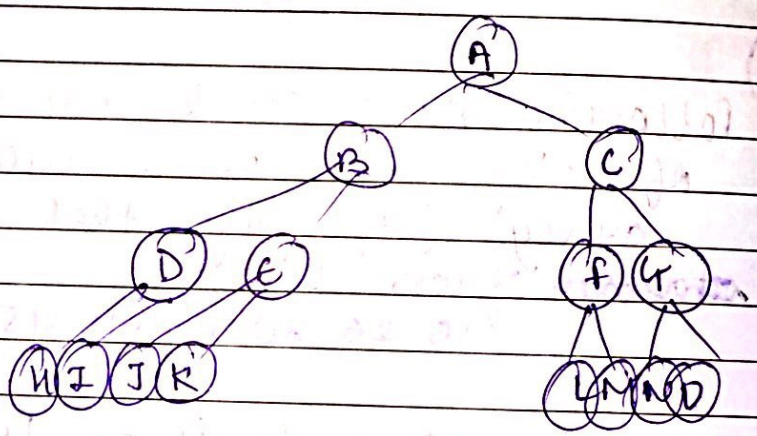
2C :- RChild (i) = $2 * i + 1$

In a link representation of binary Tree, if there are n. no. of node then the no. of null links

$$x = n + 1$$

* Traversal.

① Inorder (Left, Root, Right)
② Pre Order (Root, left, Right)
③ Post (Left, ~~root~~, ~~Right~~) right, root



Inorder:

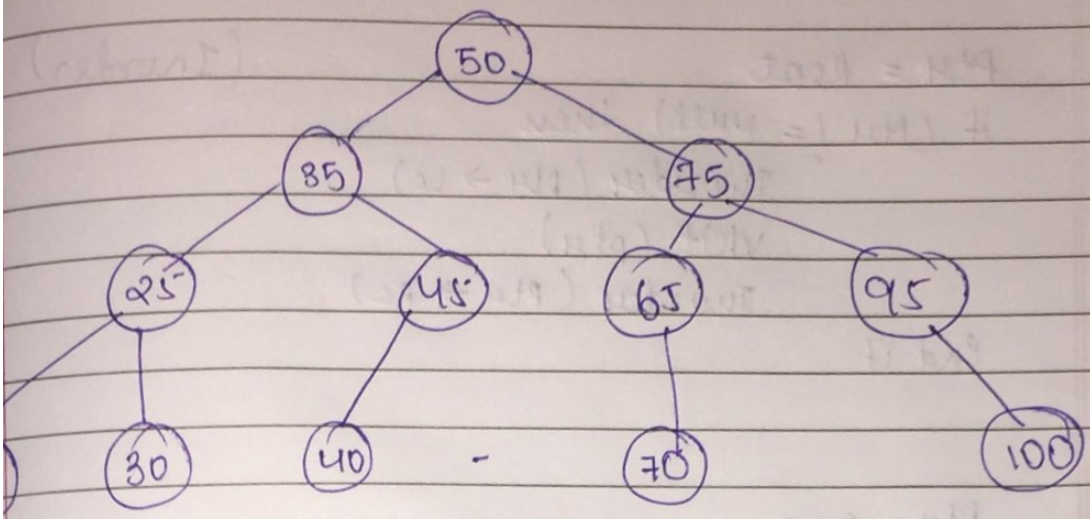
HDI B JE K A L F M C N G O

Pre Order:

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

Post Order

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



Inorder:

15 25 30 35 40 45 50 70 65 75 95 100

Pre order:

50 35 25 15 30 45 40 75 65 70 95 100

Post Order:

15 30 25 40 45 35 70 65 100 95 75 50

(1) (Inorder)

```
Ptr = Root
if (Ptr != Null) then
        Inorder (ptr -> LC)
        visit (ptr)
        Inorder (Ptr -> RC)
End if
```

(2) (Pre)

```
Ptr = Root
If (ptr != Null) then
        visit (ptr)
        Pre Order (ptr -> LC)
        Pre Order (Ptr -> RC)
end if
```

(3) (Post)

```
Ptr = Root
  if (Ptr != Null) then
        Postorder (Ptr -> LC)
        Postorder (Ptr -> RC)
        visit (ptr)
  end if
```