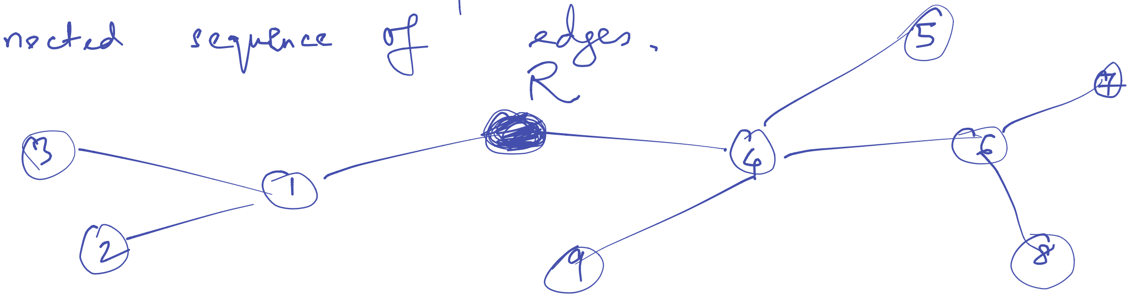
TREES

Rooted Trees

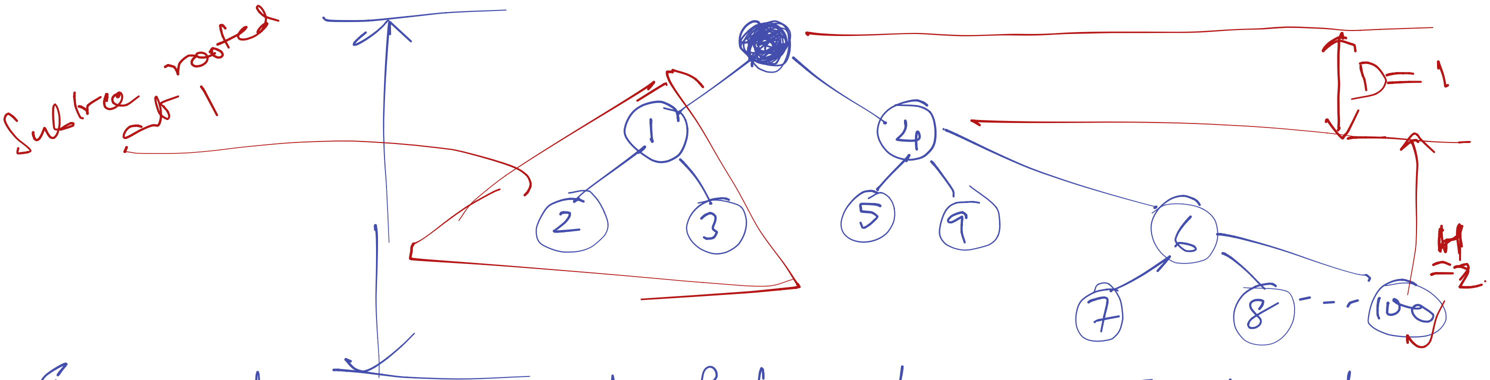
Tree : Set of nodes and edges that connect them.

→ Exactly one path between two nodes.

Path : Connected sequence of edges.



Rooted tree : One distinguished node is called the Root



Every node n, except Root, has ONE parent, p; the first node on the path from n to the root..

n is p's child.

Root has NO parent.

A node can have ANY number of children.

Leaf : Node with NO children.

Siblings : Nodes with the same parent.

Ancestors ; of a node n are the nodes on the path from n to the root (including n itself and the root)

If a is an ancestor of n, n is a descendent of a.

Length of a path : Number of edges on the path.

Node to itself → Path length = 0

Depth of node n : Length of the path from n to the Root. Depth of the Root = 0.

Height of node n : Length of path from n to its deepest descendent. Height of a leaf node = 0.

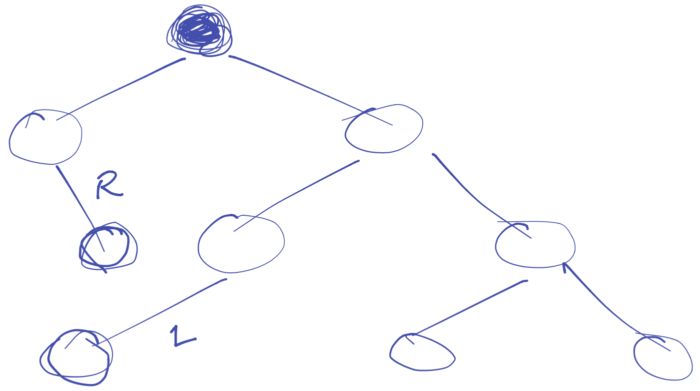
Height of a tree = Height of the Root

Subtree rooted at n : Tree formed by n and its descendents.

Binary tree : No node has >2 children

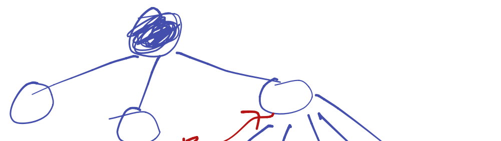
↳ ≤ 2 children

Every child is either the left or the right child (even if it is the ONLY child)



Representation of Rooted trees

- Item
- Parent
- Siblings are directly linked.

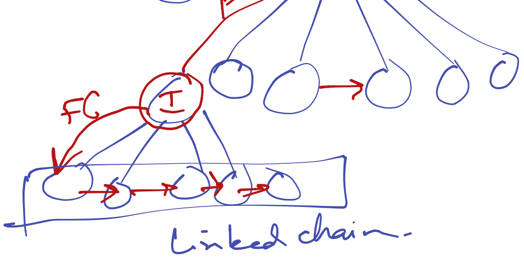


```
class   TreeNode {

        Object   item ;

        TreeNode *   parent ;
        TreeNode *   firstChild ;    } Linked
        TreeNode *   next Sibling ;  }  chain .
 }

class RootedTree {
        TreeNode *   Root ;
}
```



Linked chain.