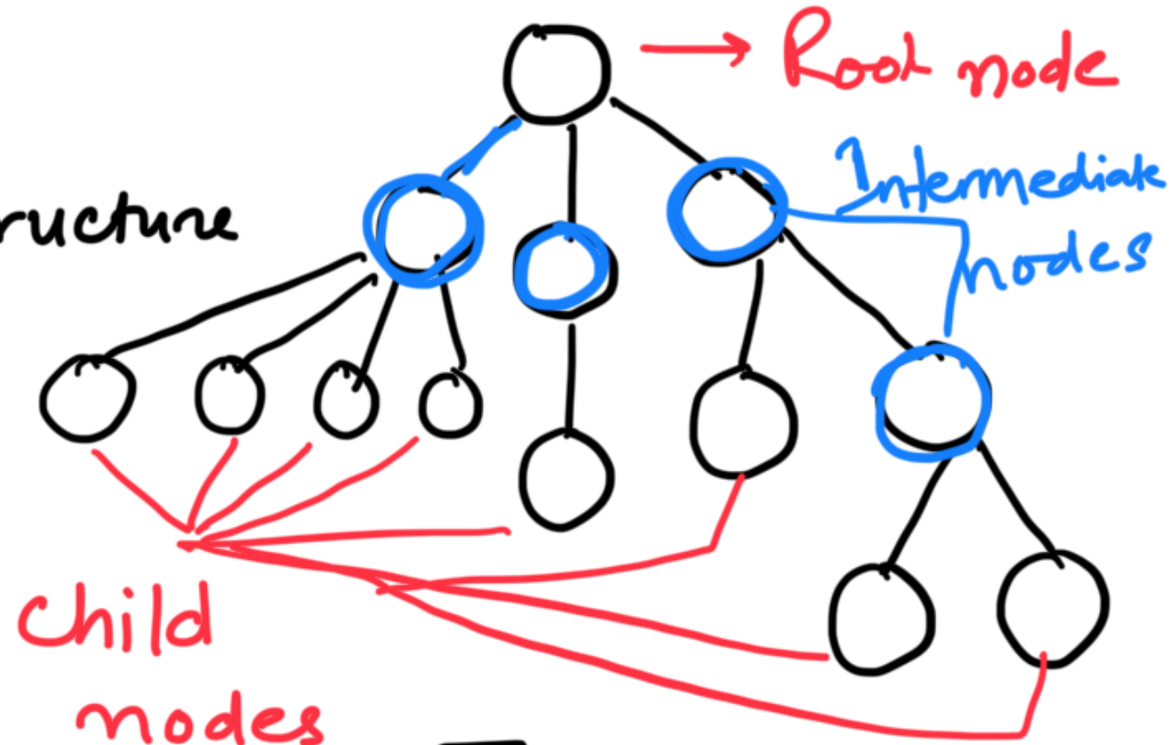
Tree -

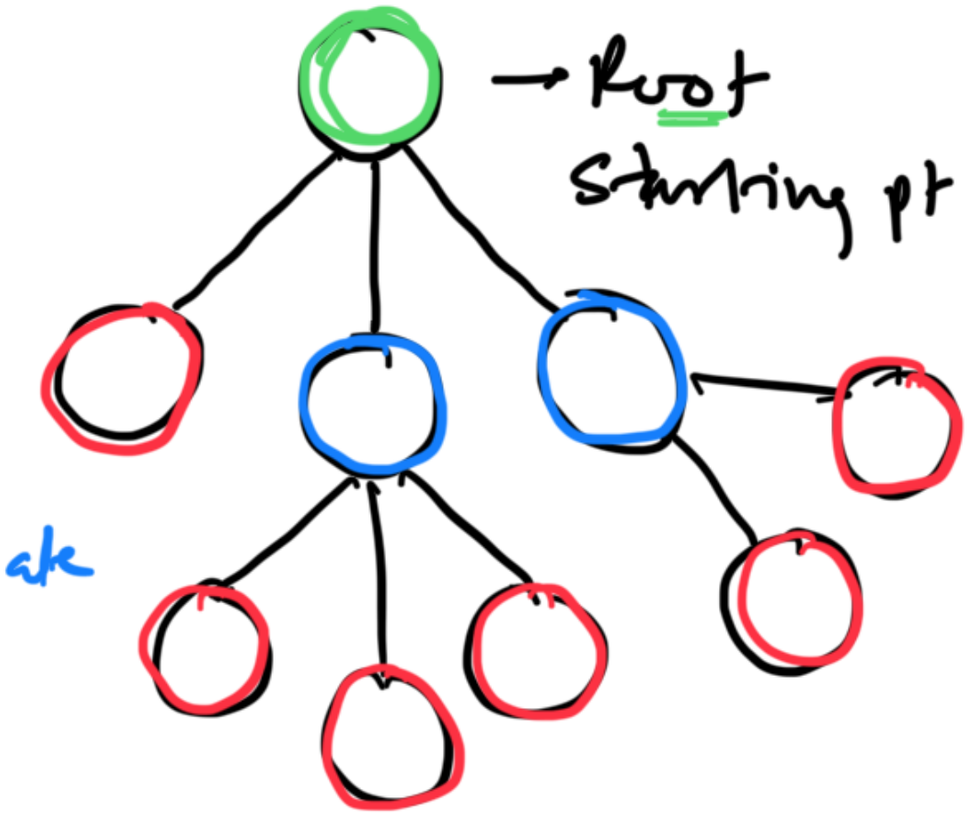
- hierarchical data structure to represent and organize data
- Nodes + Edges



Tree

——

- Parent - child



Type of structure defines the different names of the child

Tree Terminologies

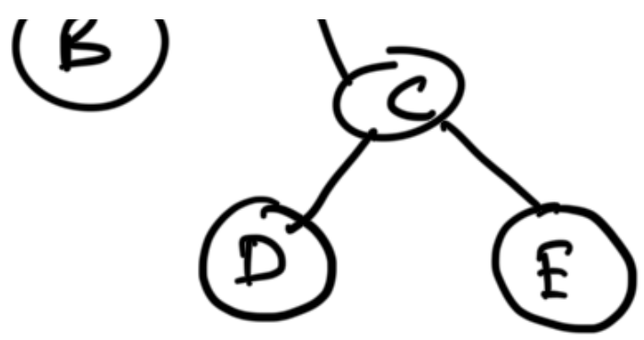
1. Parent Node — node that has 1/mor child nods
2. Child Node - Node that is a direct descendant of another node (descendant)
3. Root node - having no parents
4. Leaf mode - having no childrens

Root node - A

Child nodes - B, C, D, E

Parents Node - A, C

Leaf node - B, D, E



5. Ancestor — predecessor in the path from the root node

E → Ancestor → A, C

6. Descendants - Successor in the path from the root

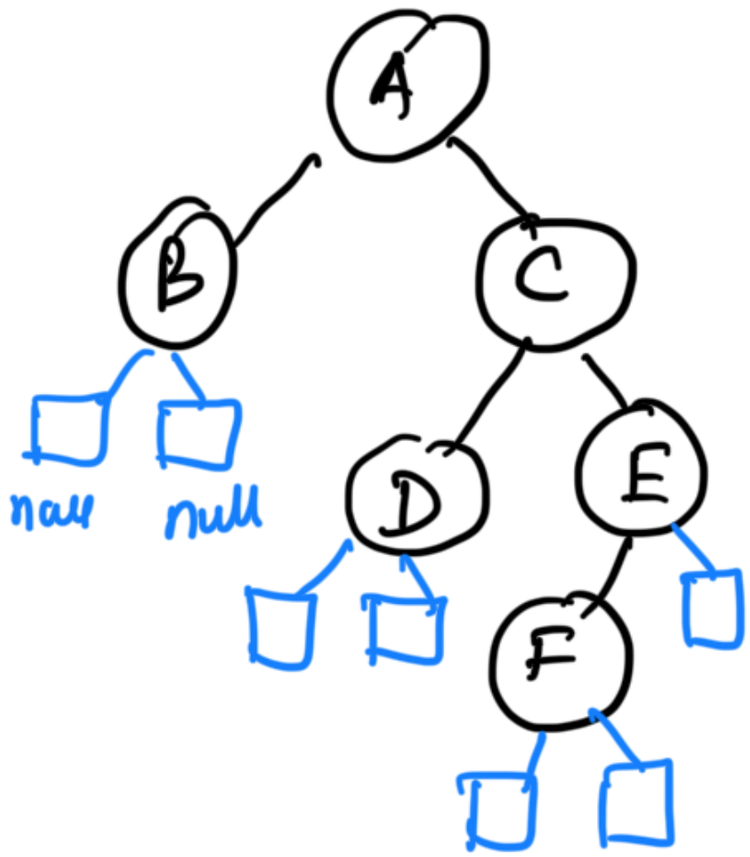
A → Descendants — B, C, D, E

7. Sibling — nodes that share the same parents.

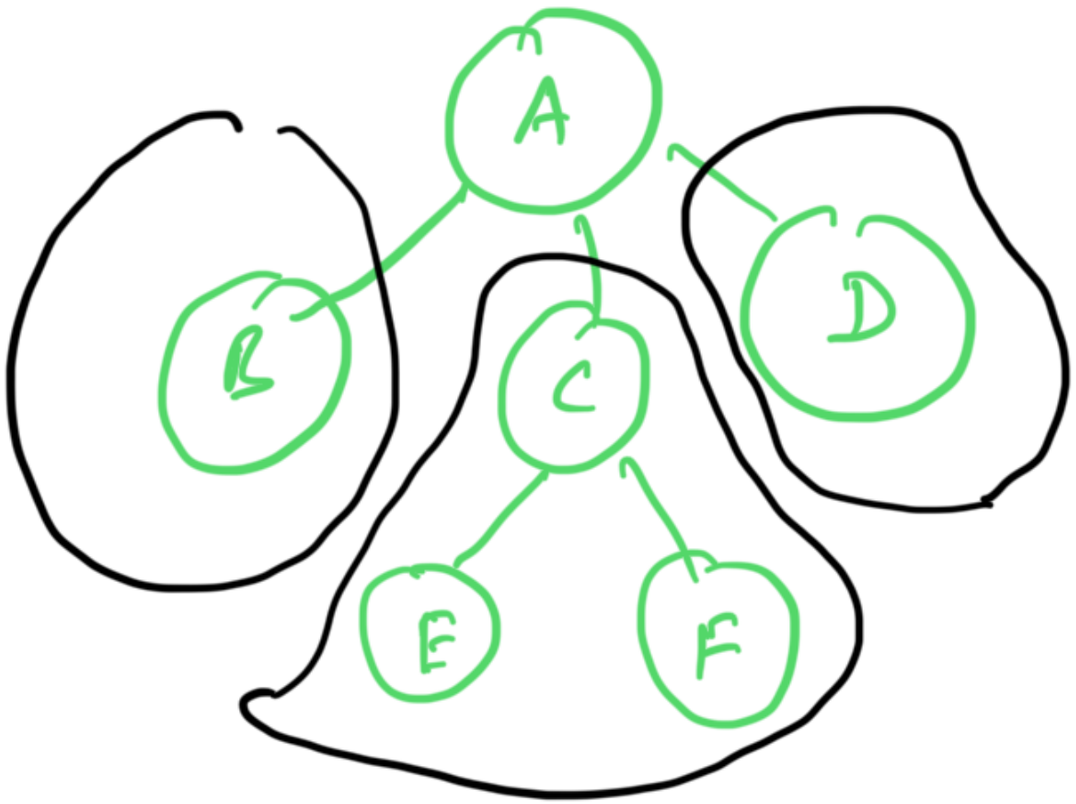
(B, C), (D, E)

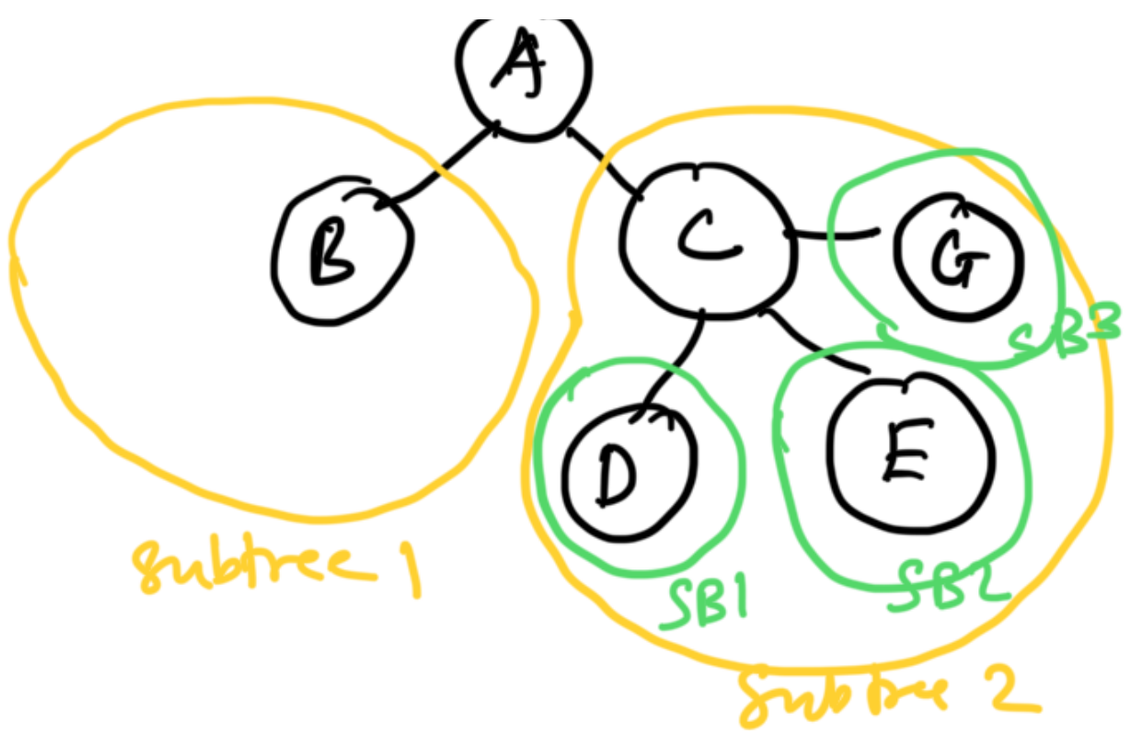
8. Internal Nodes → Nodes with atleast one children is called as Internal nodes.

Internal nodes } A, C, E



9. External nodes → Null Nodes

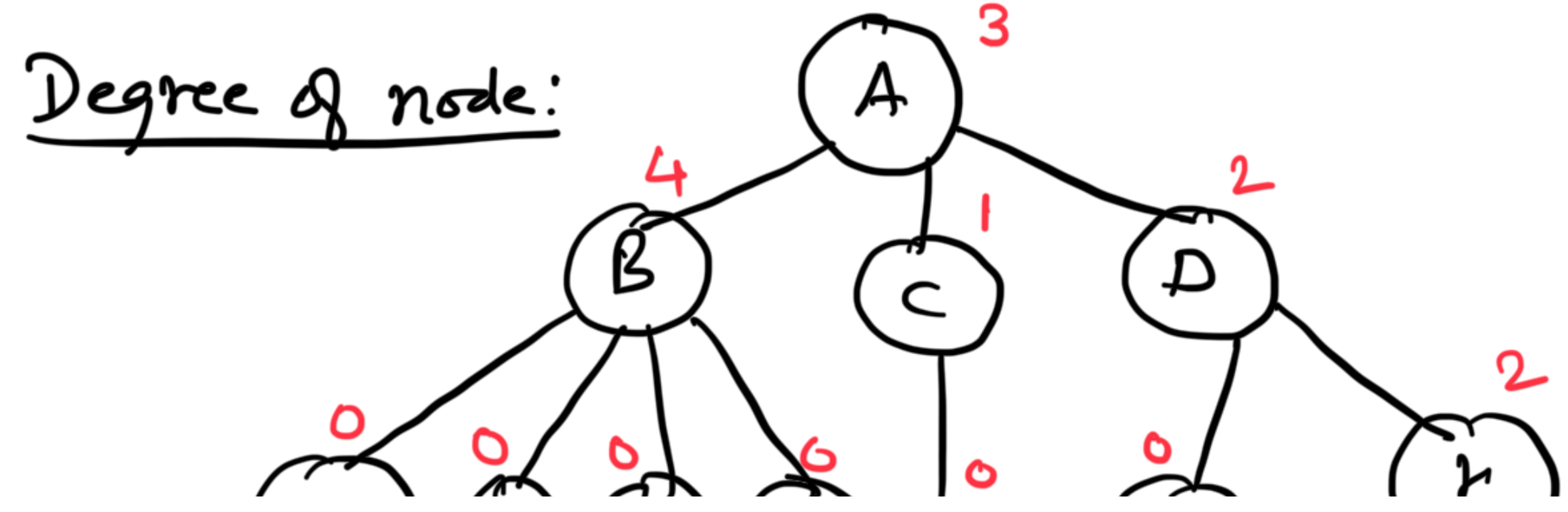


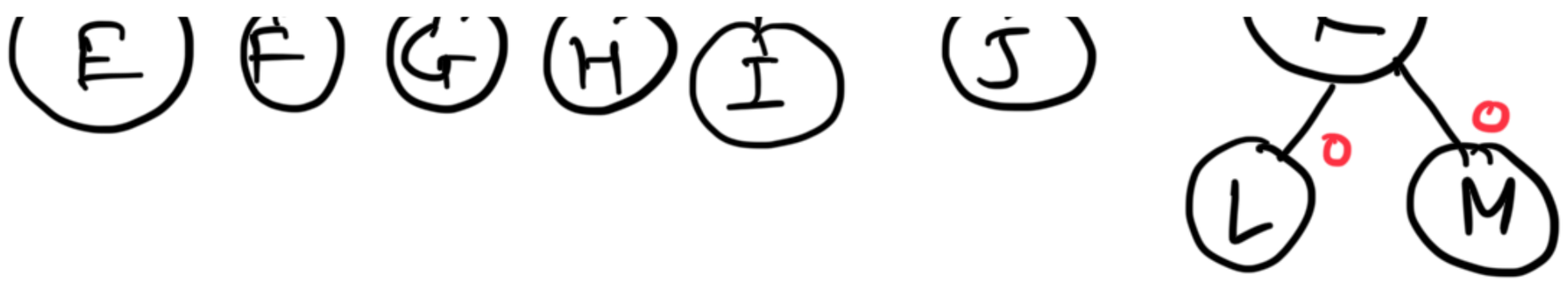


forest

Forest

Degree of node:





Leaf Nodes = Degree = 0

Degree (A) = 3

Degree (B) = 4

Degree of Tree

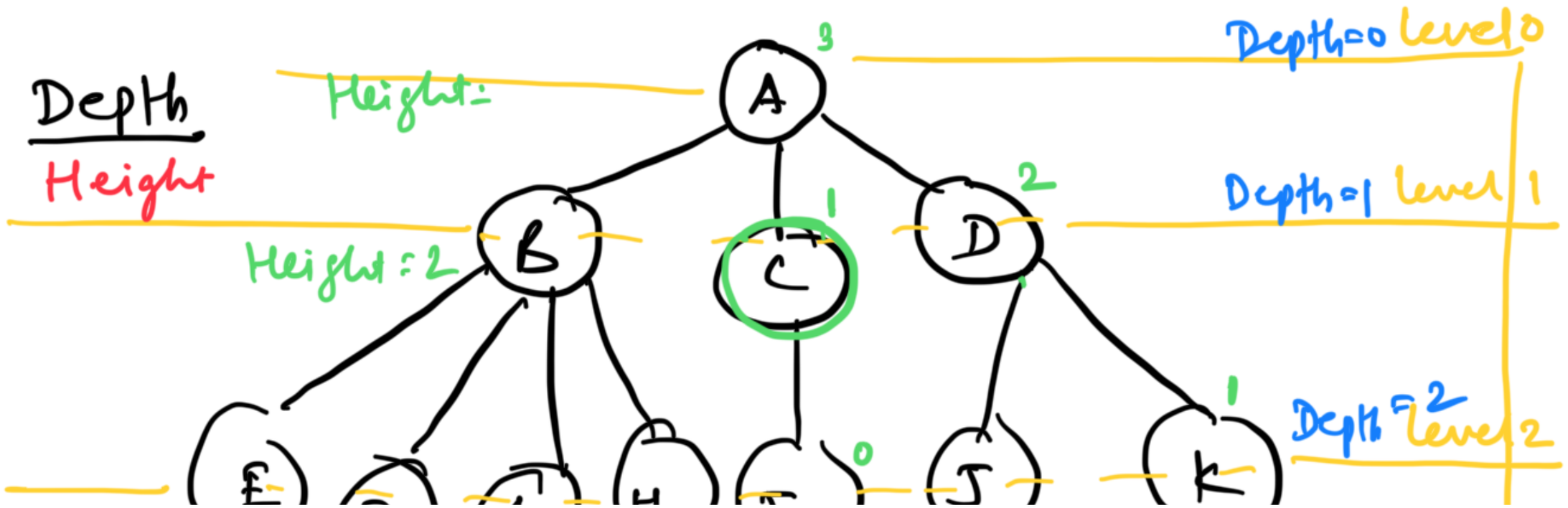
Highest degree in Nodes

↓

4

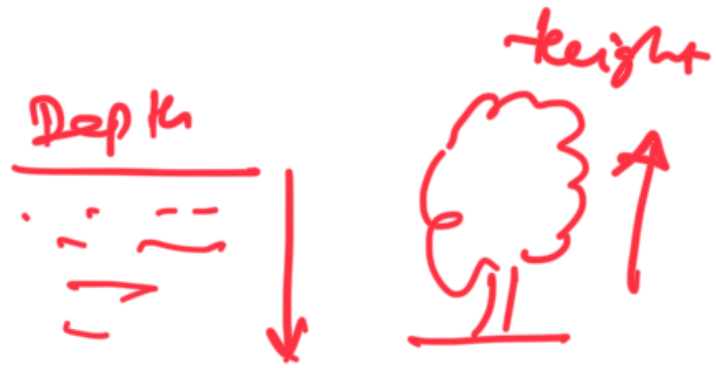
Depth

Height



Height (C) = 1 | Depth (K) = 2

Height (A) = 3 |



Level - The root of a tree is at level 0 and the nodes whose parents is root at level 1

Height - The height of a nodes is the number of nodes on the longest path from the node to a leaf node.

Properties of Tree:

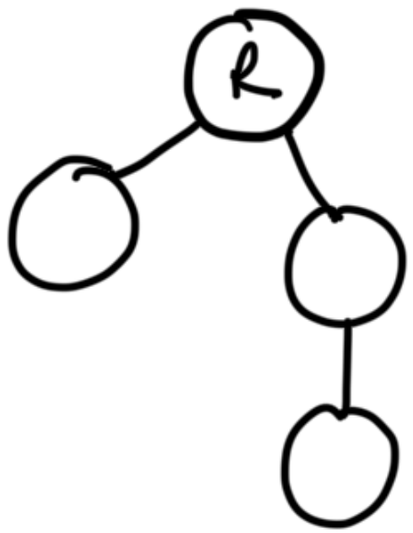


fig 1 : Tree

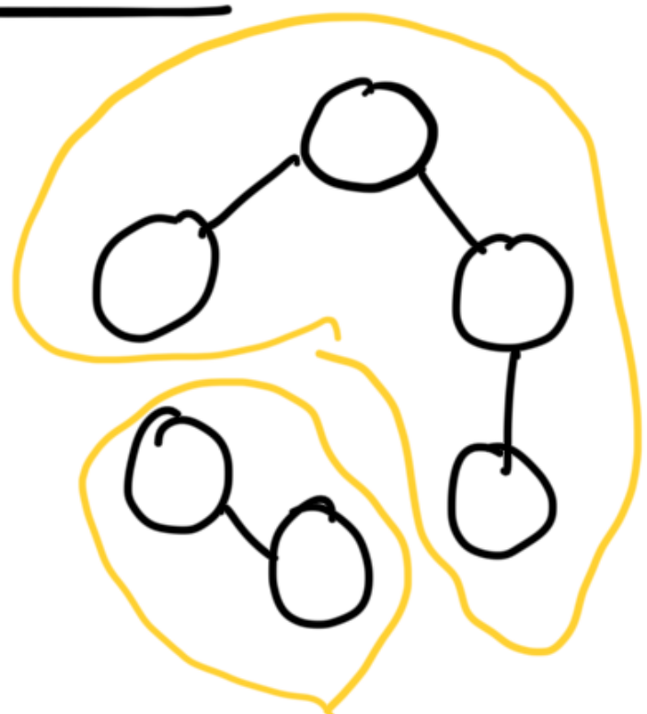


fig 2

Not a tree

All nodes required to be connected

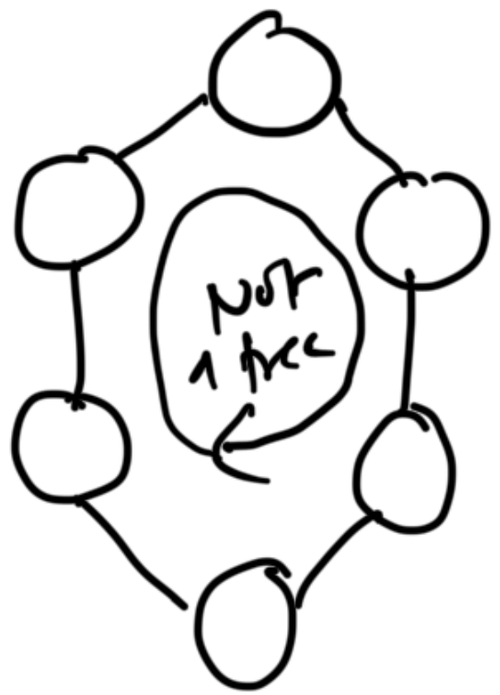
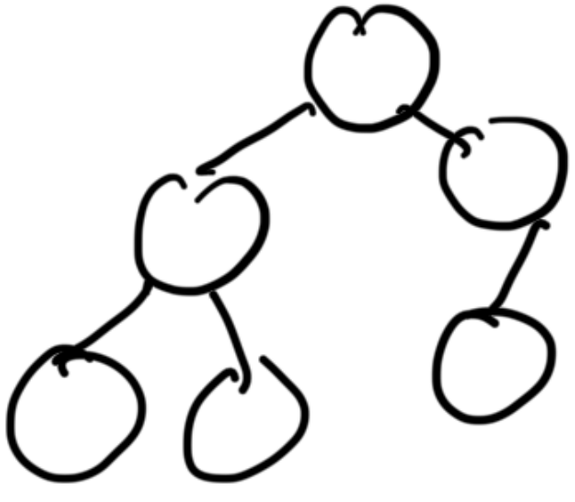


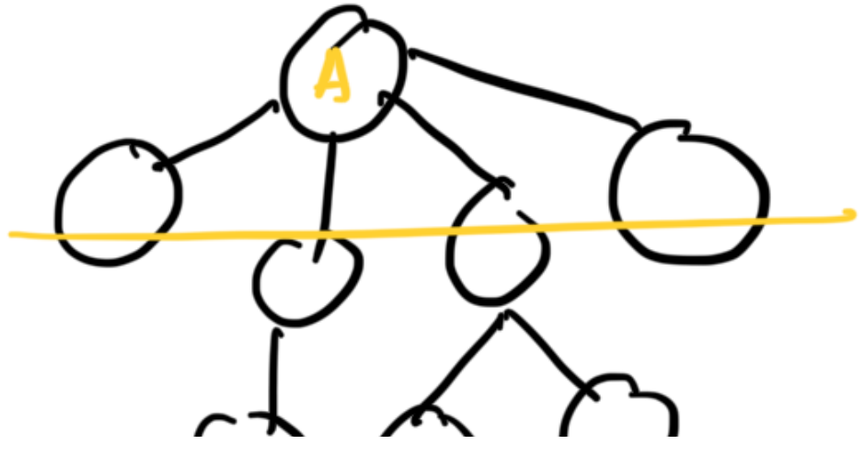
Fig 3

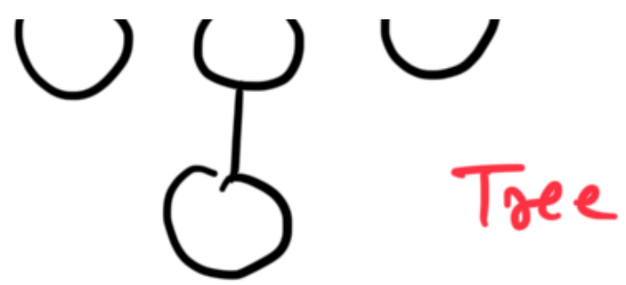
Not a tree

it contains cycle



Tree



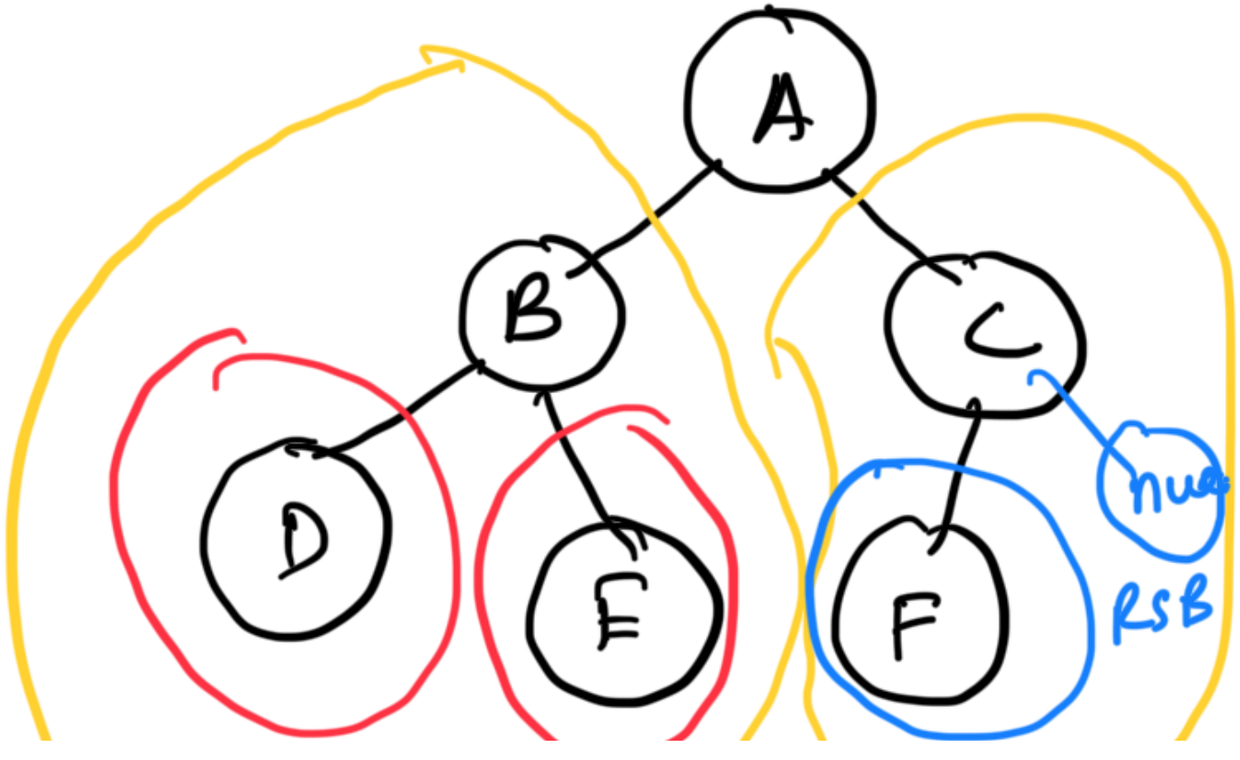


Tree

Binary Tree : It is a tree in which every node has **at most 2** children

$\{0, 1, 2\}$

No of childrens = $\{0, 1, 2\}$



Partition creates only 2 subpaces

1) Left subtree

2) Right subtree


LST
RST
USB
Left subtree
Right sid subtree