CS & IT ENGINEERING

Data Structure

Tree
Chapter- 5
Lec- 01

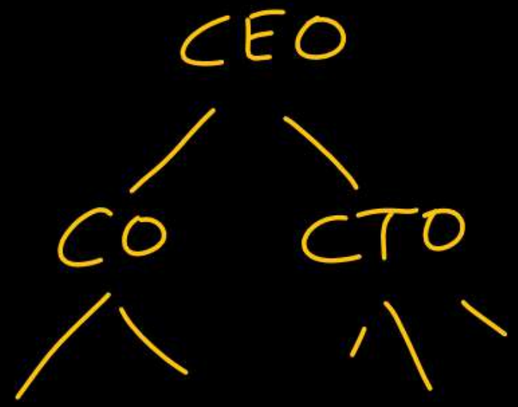
By- Pankaj Sharma sir

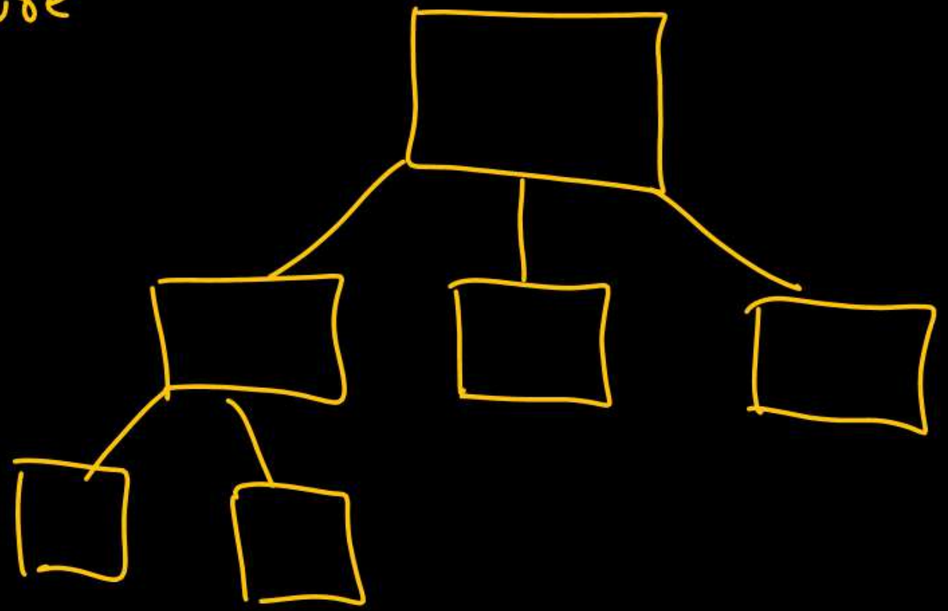
Tree-I

Trees

Non Linear data structure

1. Organization structure
2. Folder structure

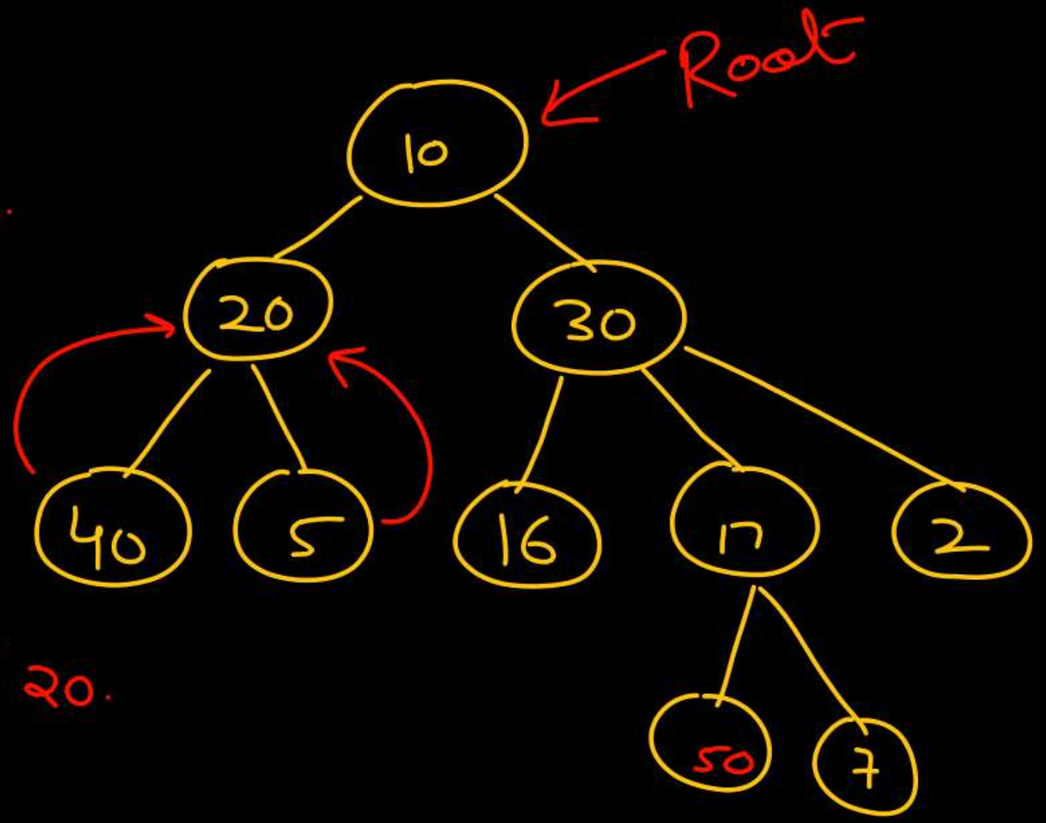




STL

3. HTML/XML (JSON Objects)
4. Binary Search trees
5. Binary Heap
6. B-Trees, B^+-Trees
7. Parse tree / Expression

Trees



① Node : Each element is rep. by node

② Child : 20, 30 are childs of 10.

40, 5 are childs of 20.

③ Parent : 20 is the parent of 40, 5

30 is the parent of 16, 17, 2

④ Root : Distinguishable from other node (No parent)

Trees

Leaf node : Node with no child.

40, 5, 16, 50, 7, 2

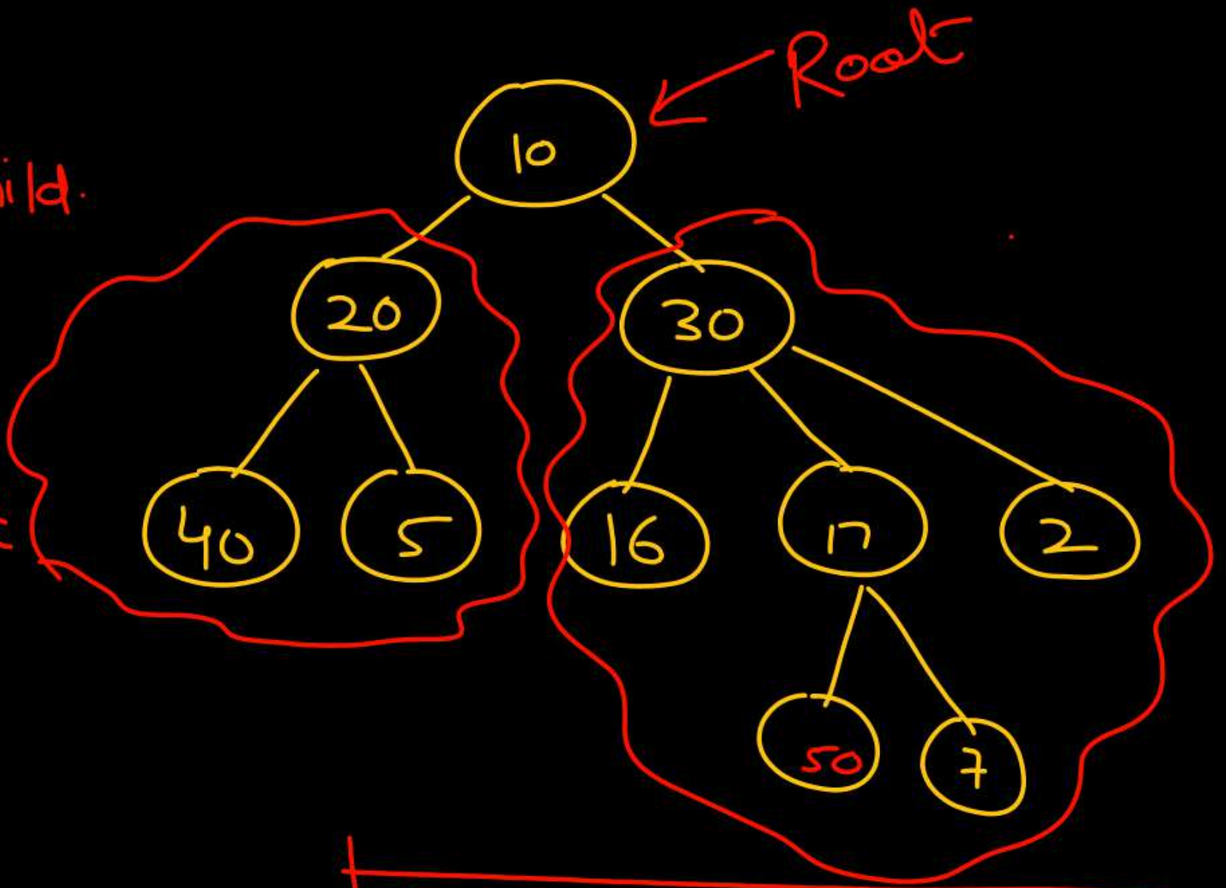
Internal node : Node with atleast 1 child.

i.e. 10, 20, 30, 17

degree of a node : No. of childs.

degree of a leaf node = 0.

degree of node with 1 child = 1



degree of node with 20 key = 2

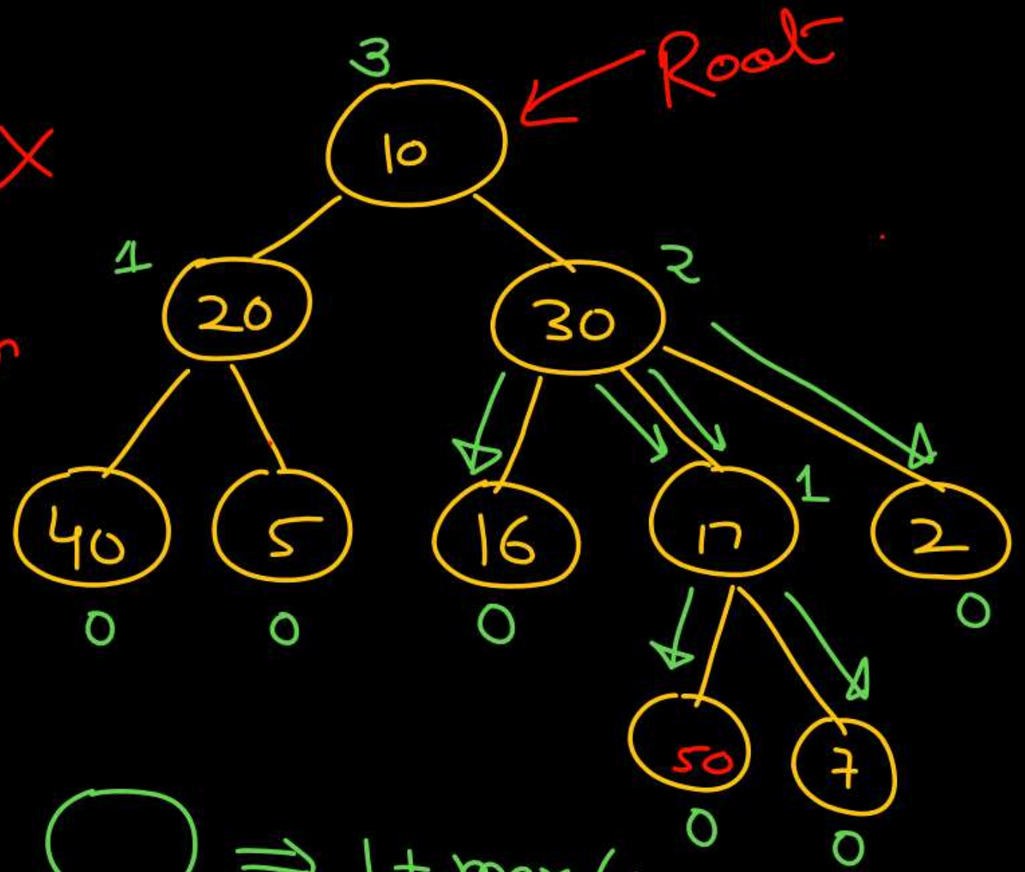
Trees

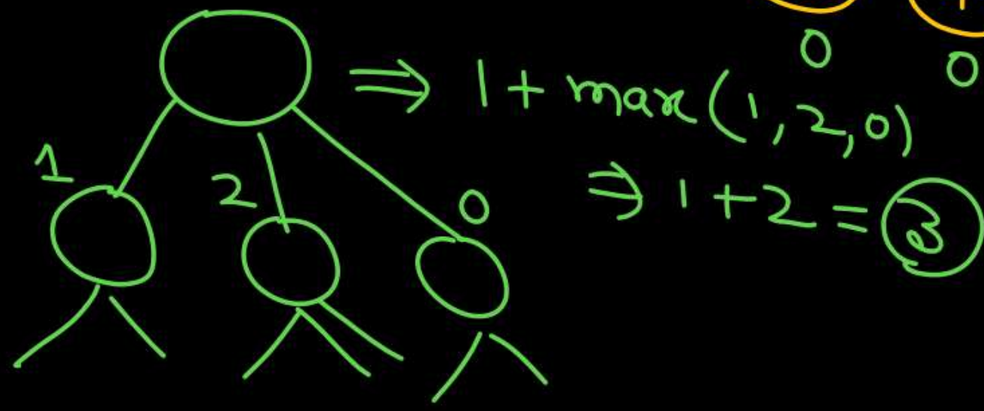
Height of a node : Height of a node X

is the length of path from node X to the farthest leaf node.

Height of a leaf node = 0

Height of tree = Height of root node



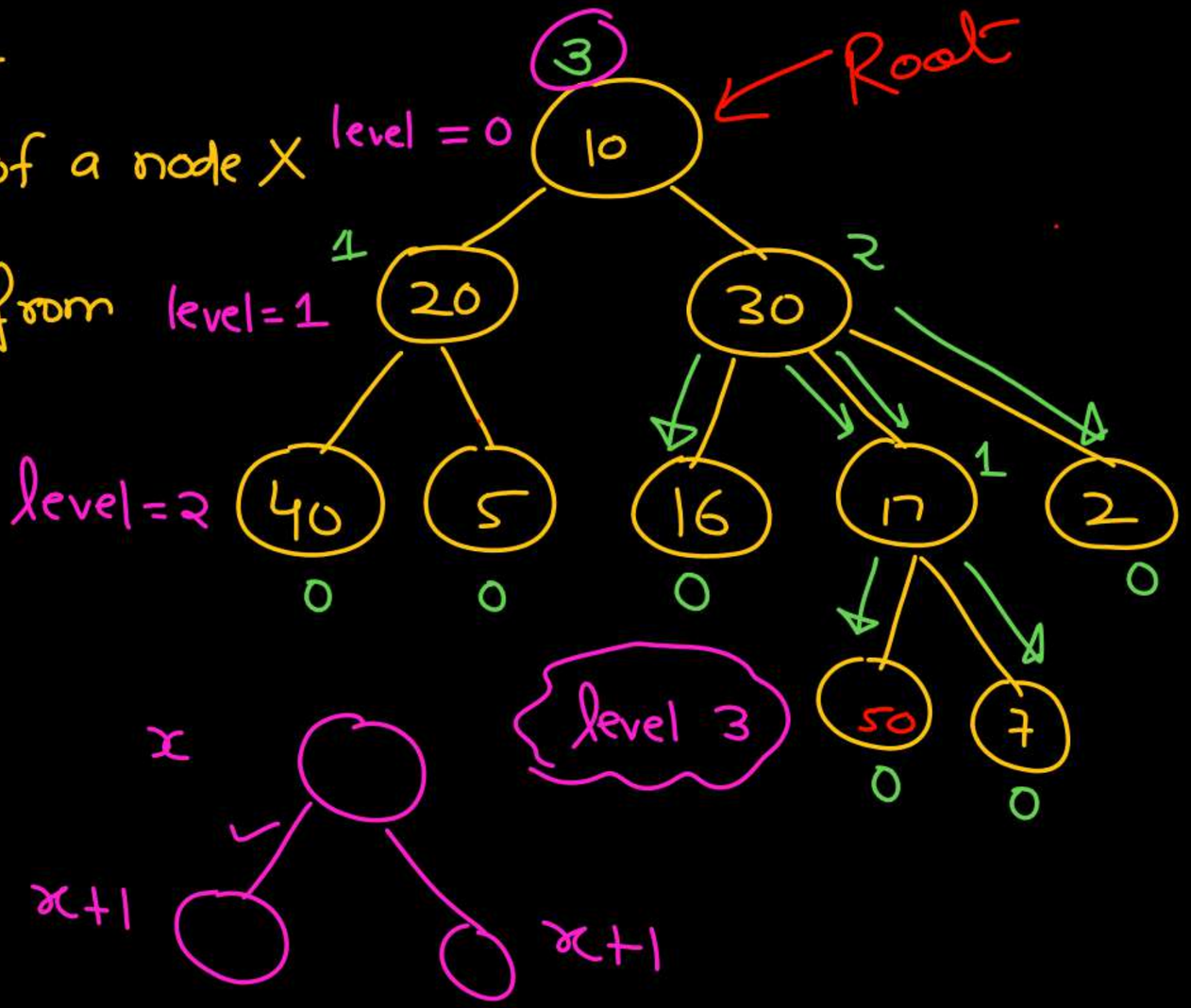


$\Rightarrow 1 + \max(1, 2, 0)$

$\Rightarrow 1 + 2 = 3$

Trees

Depth/level of a node : level of a node X is the length of path from root node to X.



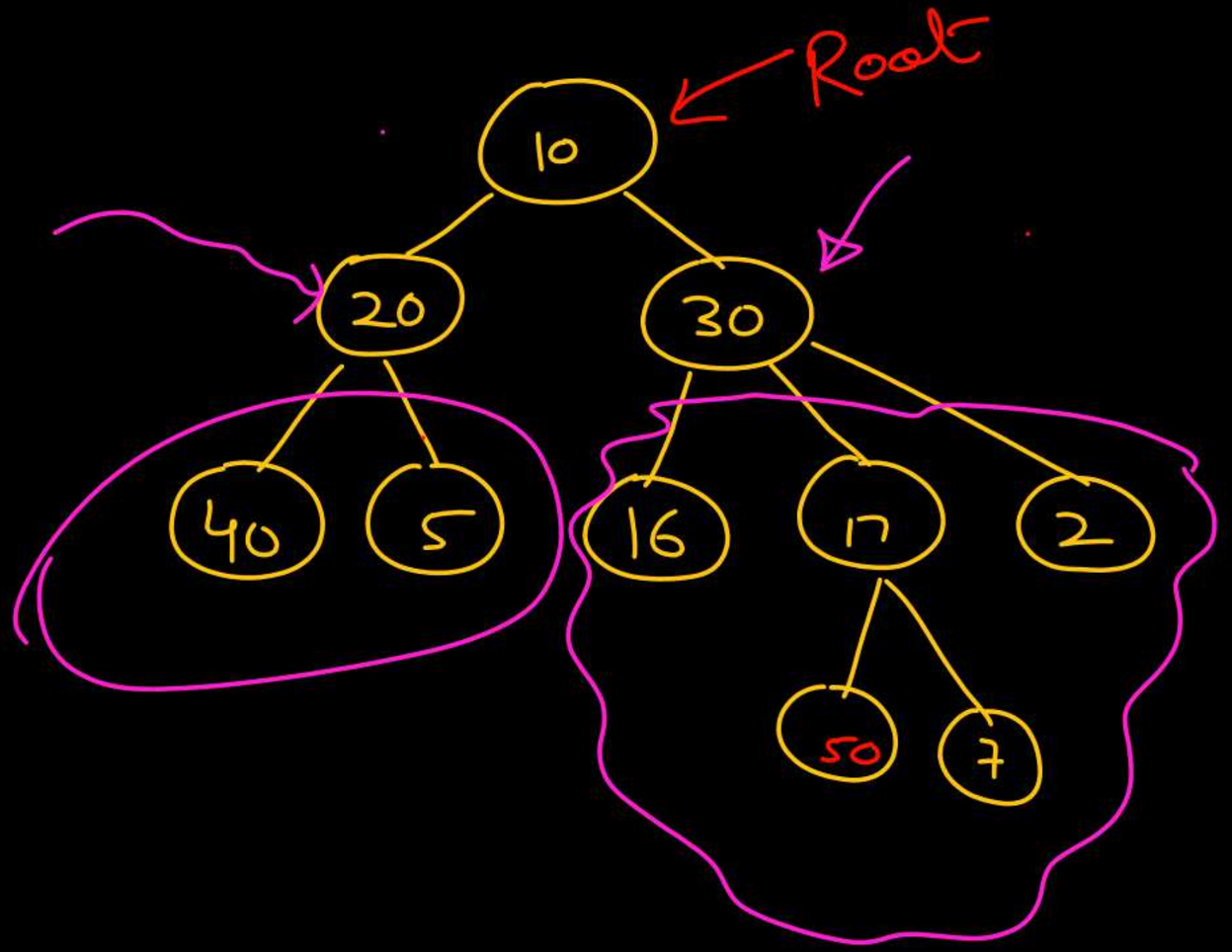
Trees

Ancestor

40 ⇒ 20, 10
5 ⇒ 20, 10
50 ⇒ 17, 30, 10
7 ⇒ 17, 30, 10
2 ⇒ 30, 10

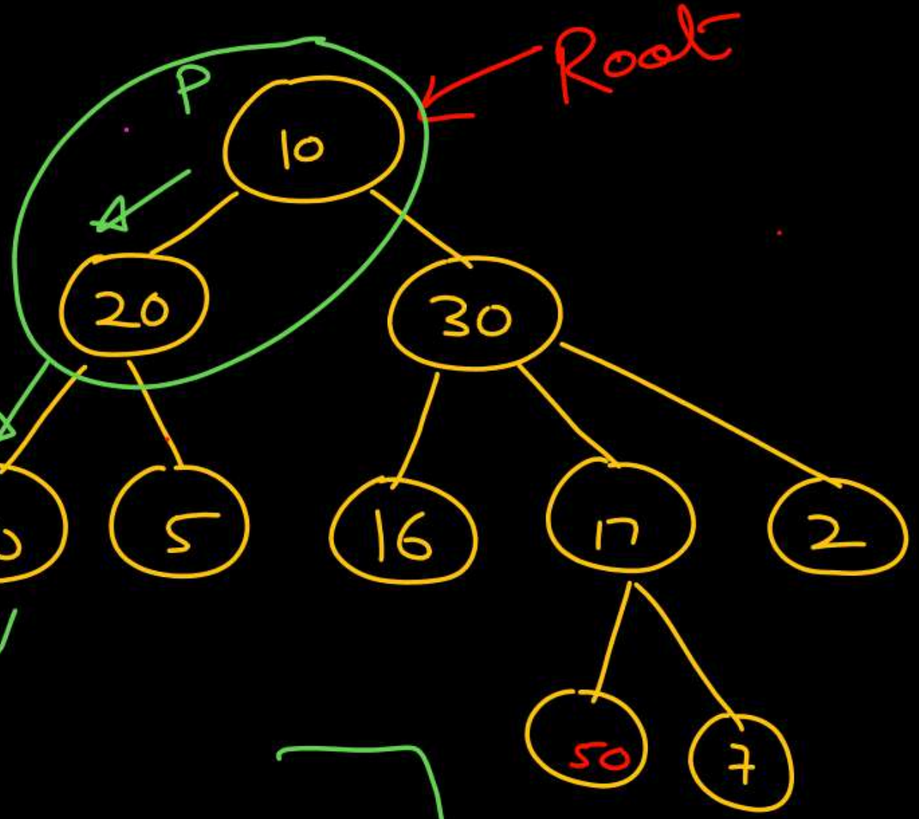
descendant

20 ⇒ 40, 5
30 ⇒ 16, 17, 2, 50, 7



Trees

Ancestor of a node q : If there is a path from node P to node q, then all the nodes in the path (other than q) are ancestor of q.



OR

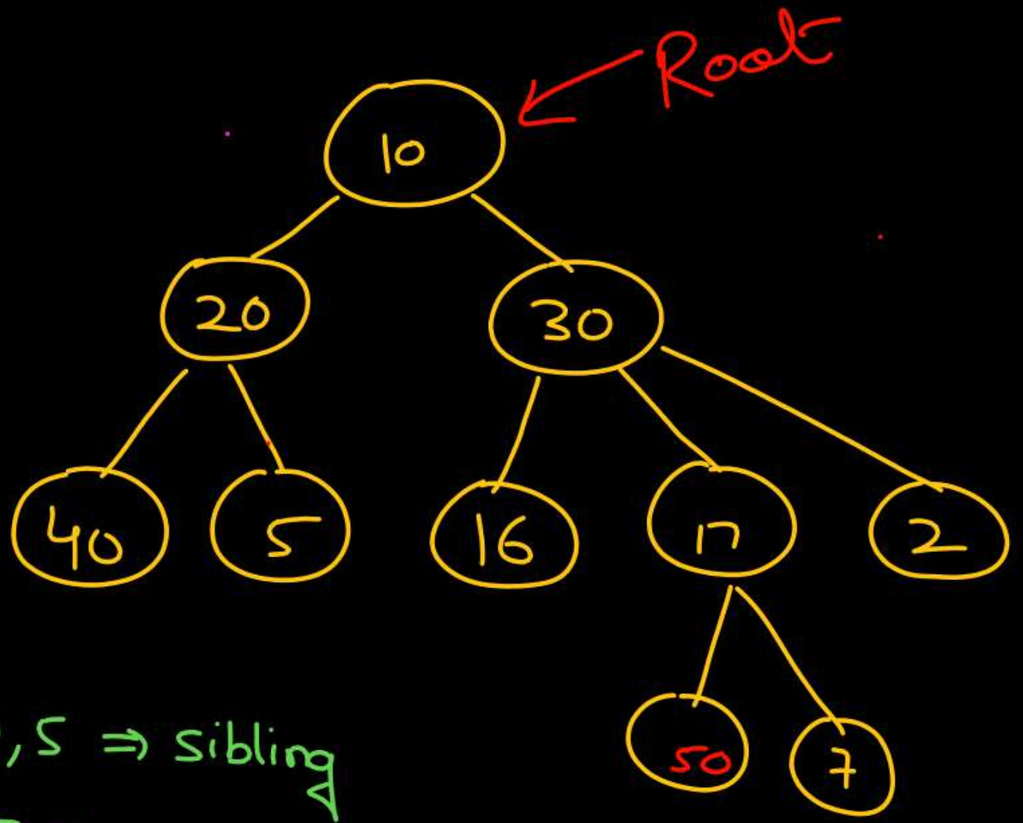
(i) Parent is the ancestor of a node

and

(ii) Parent of some ancestor is also an ancestor

Trees

Sibling : Nodes having same parent



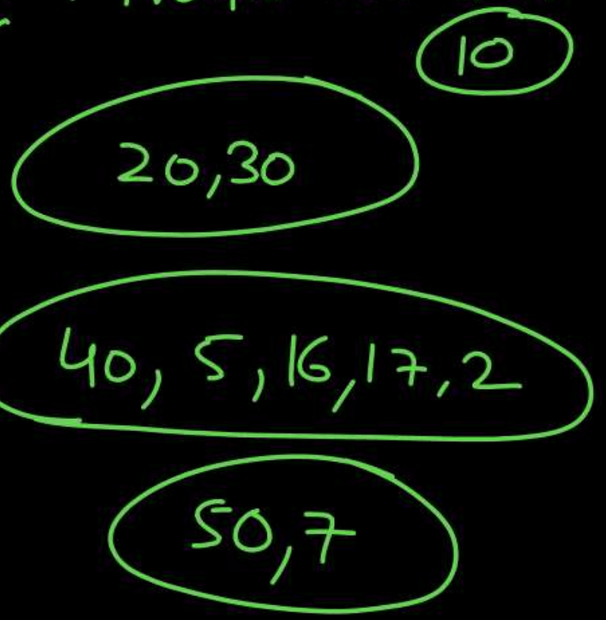
40, 5 ⇒ sibling

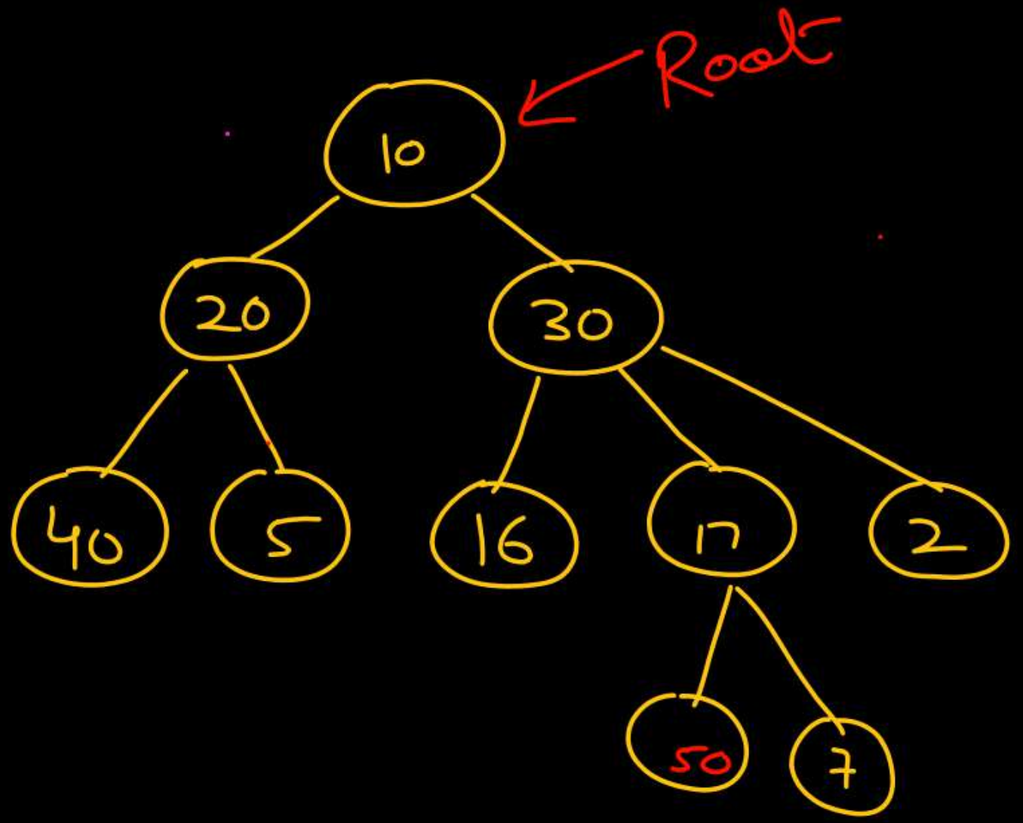
16, 17, 2 ⇒ sibling

20, 30 ⇒ sibling

Trees

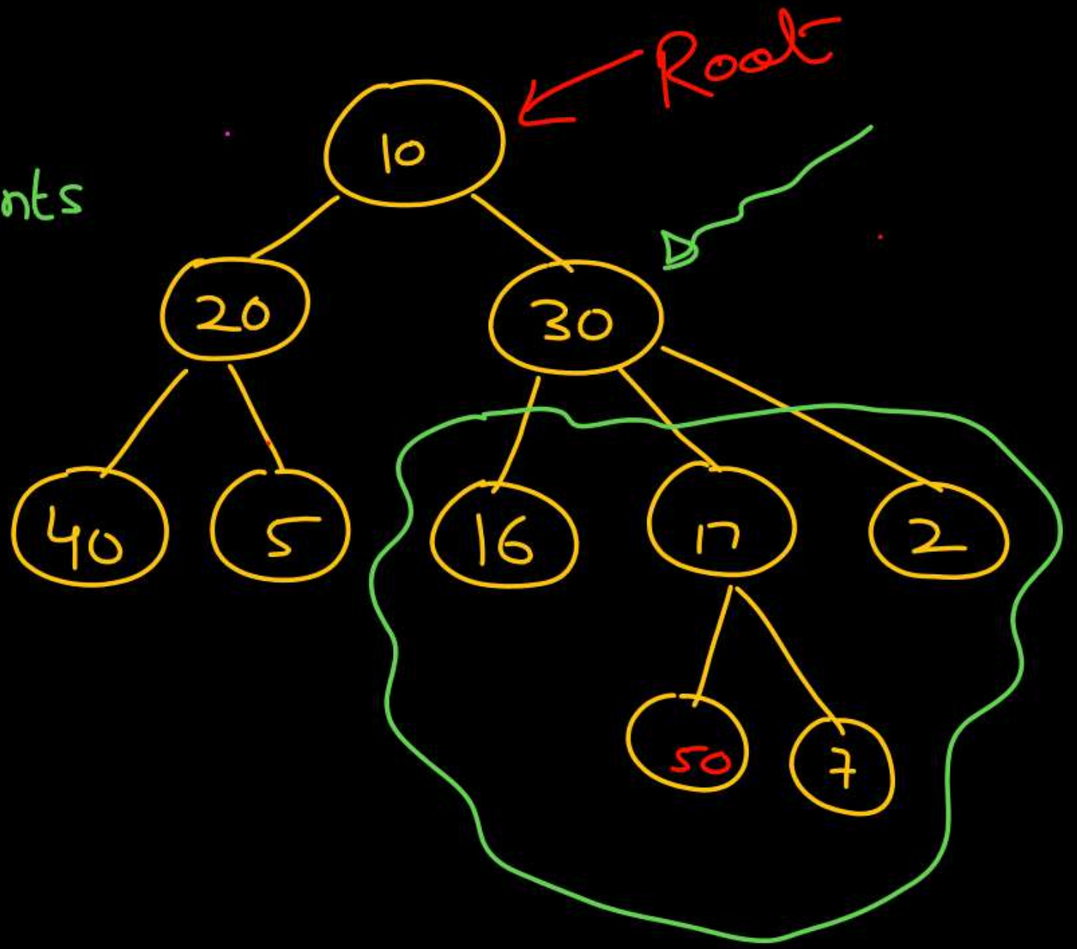
Generation : all Nodes at a level





Trees

Size of a node: No. of descendants of the node (including the node itself)



Binary tree

Directory

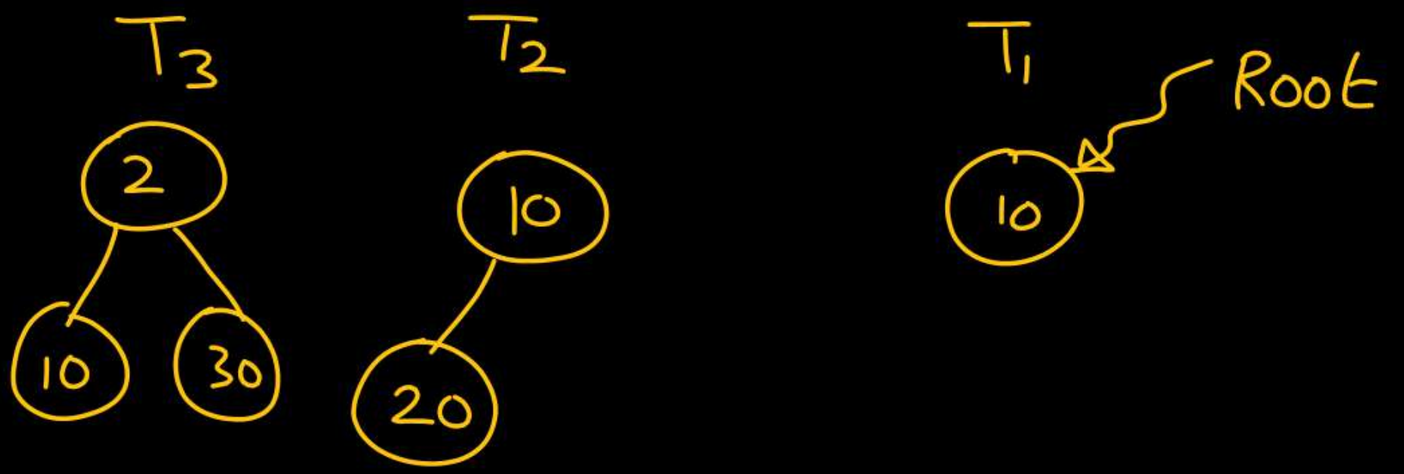
A node can have almost 2 child.

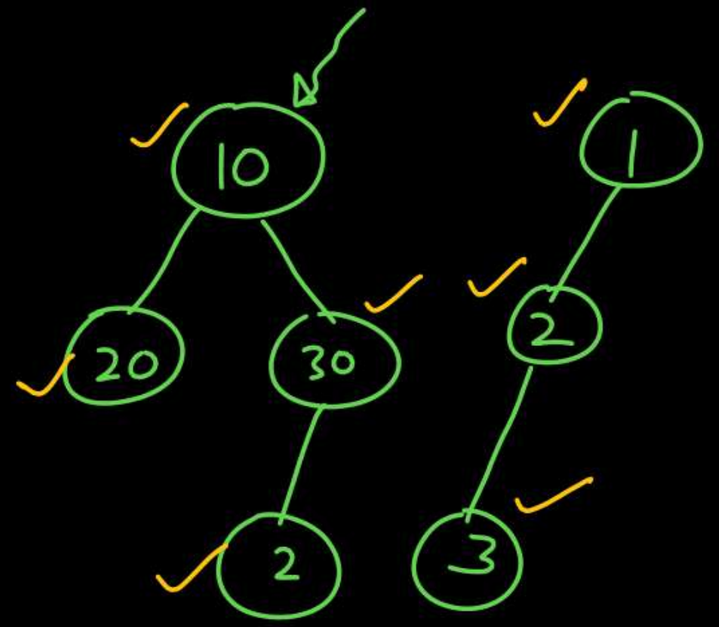
0 child ⇒ leaf

1 - child

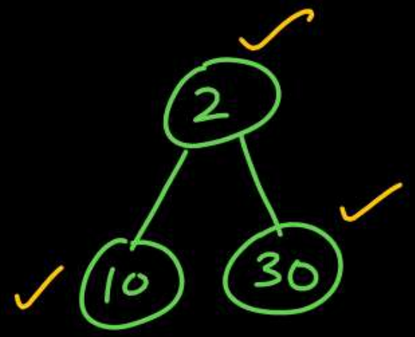
2 - child

] Internal node

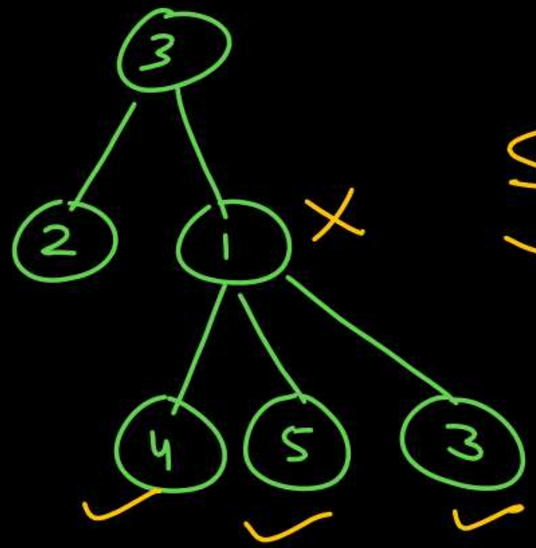




Binary tree

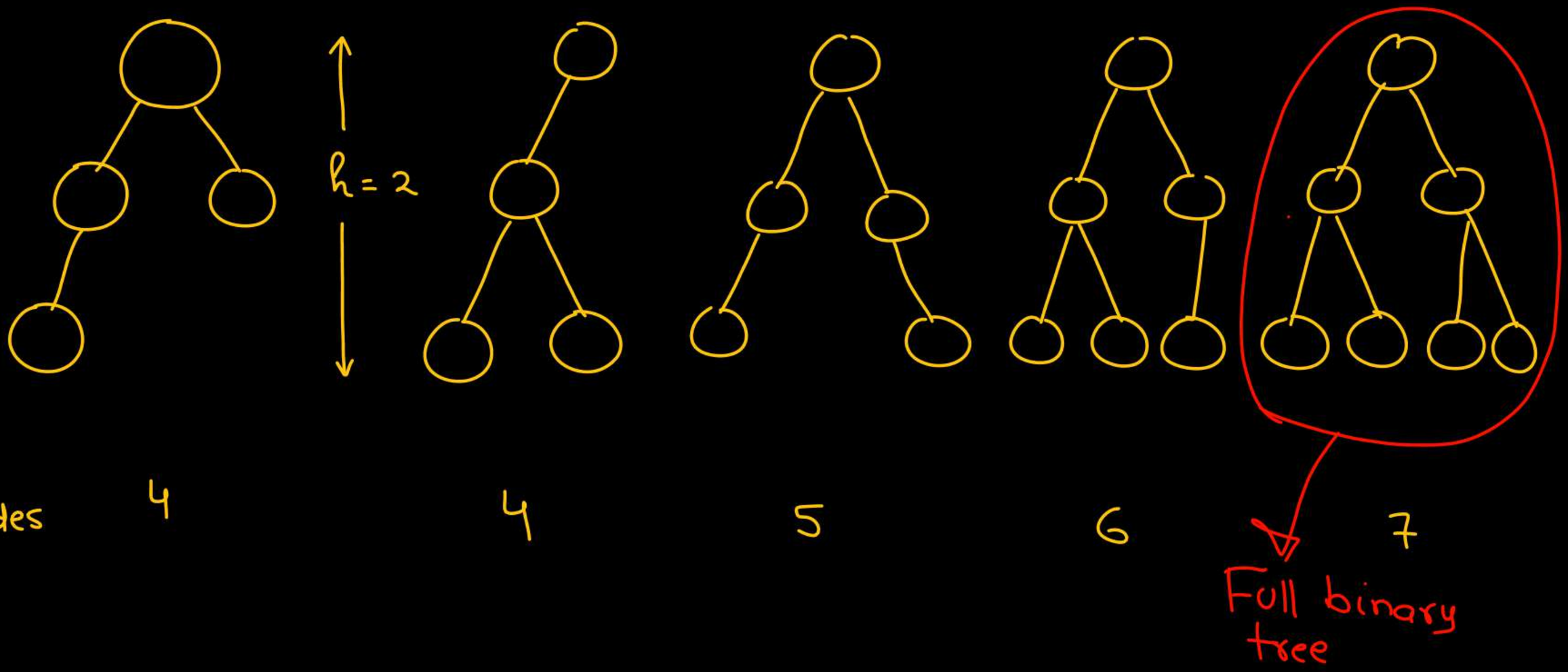


Binary tree



Not a binary tree

Sirsha


h = 2
# nodes
4
4
5
6
7
Full binary tree

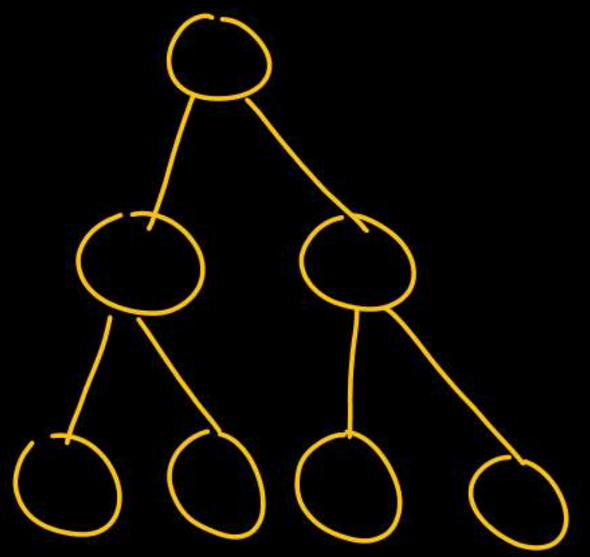
Full binary tree

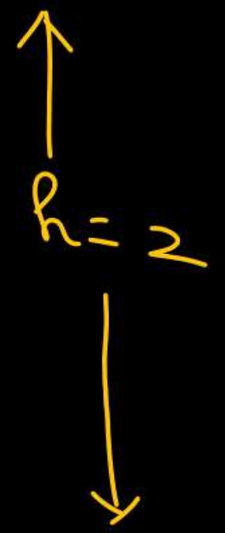
Max. no. of nodes in a binary tree of height h ?

1 ← level 0

2^1 ← level 1

2^2 ← level 2





$\#\text{nodes} = 1 + 2^1 + 2^2 = 7$

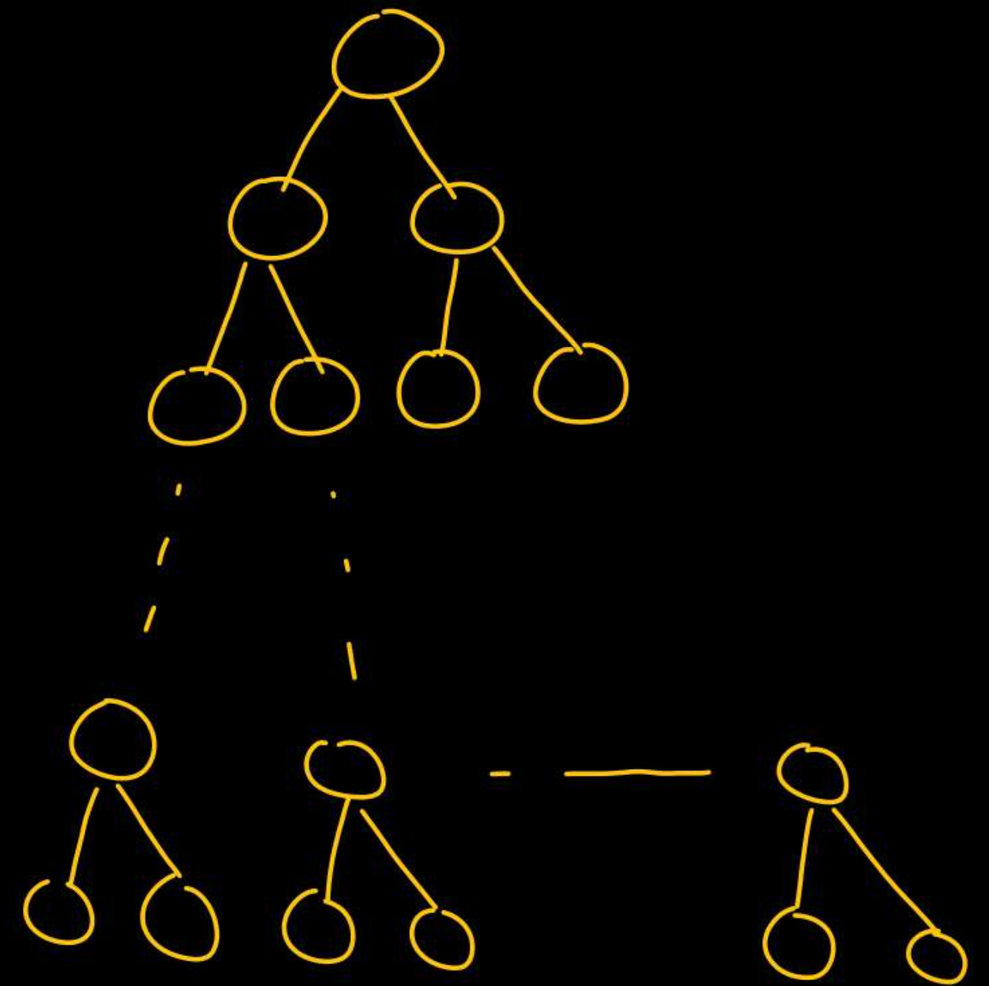
#nodes	level
1	0
2^1	1
2^2	2
2^{h-1}	$h-1$
2^h	h

Total :- $1+2^1+2^2 + \cdots 2^{h-1}+2^h$

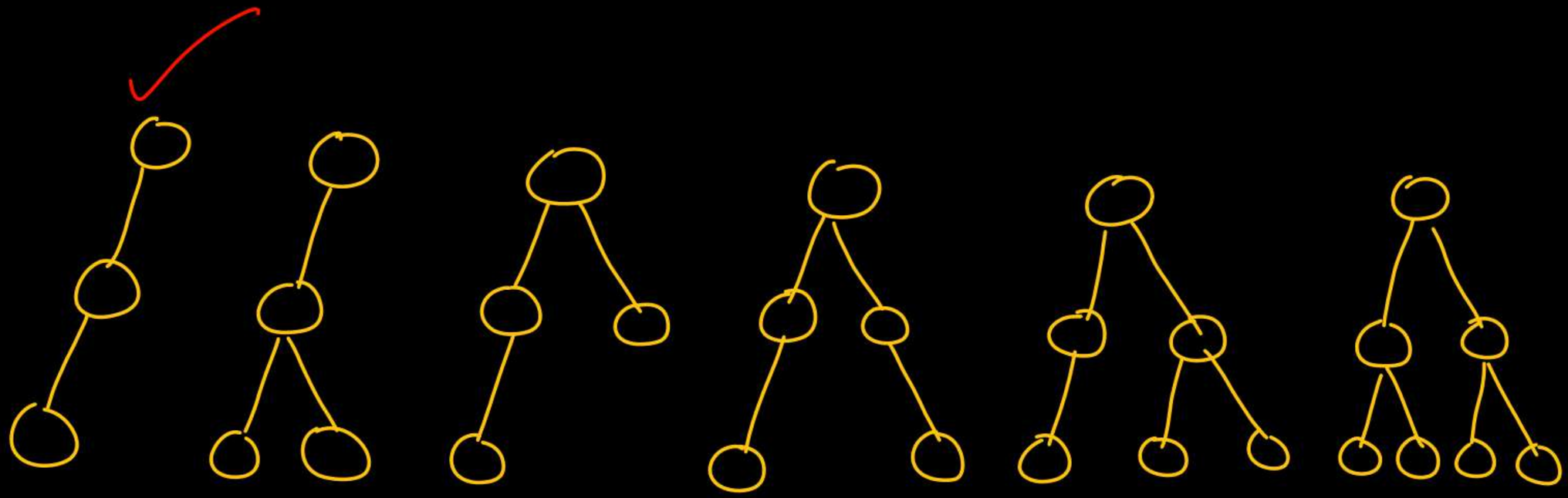
$\Rightarrow \underbrace{2^0+2^1+\cdots+2^h}_{h+1 \text{ terms}}$

$\Rightarrow \frac{1(2^{h+1}-1)}{2-1}$

$N \Rightarrow 2^{h+1}-1$

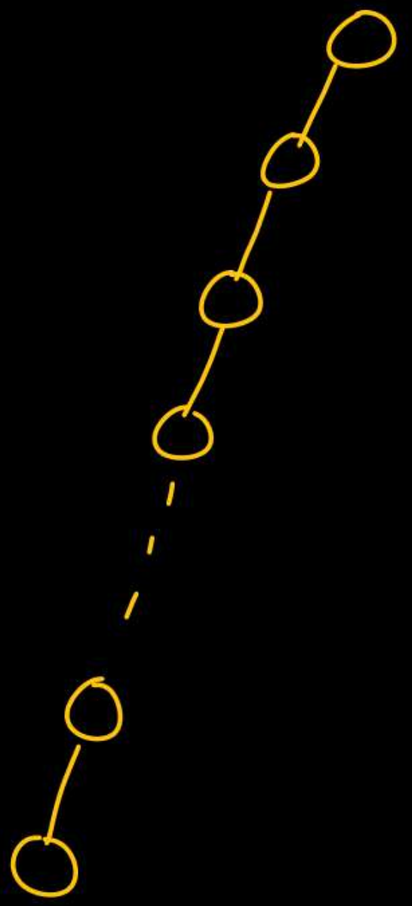


Min. no. of nodes in a binary-tree of height h?



$n = 3$

#node	level
1	0
1	1
1	2
1	3
.	.
.	.
1	h-1
1	h



$$n_{max} = 2^{h+1} - 1$$

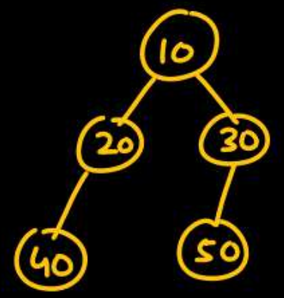
$$\Rightarrow \quad n_{min} = h+1$$

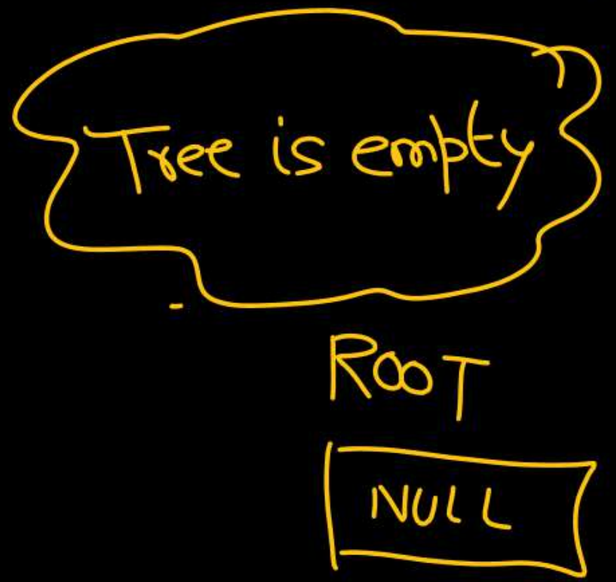
Total = $1 + 1 + 1 + 1 + \cdots$ $(h+1)$ times

5 node ⇒ 6 NULL

```
struct node{
struct node *Left ;
    int data ;
struct node *Right ;
};
```

Struct node* ROOT;






THANK YOU GW
SOLDIERS !