



CS & IT ENGINEERING

Data Structures

Tree

Lecture No.- 01

By- Pankaj Sharma Sir

Recap of Previous Lecture



Topic

Stack and Queues Part - 06



Queue
✓
PQ
PYQS

Topics to be Covered



Topic

Tree

basic terminology



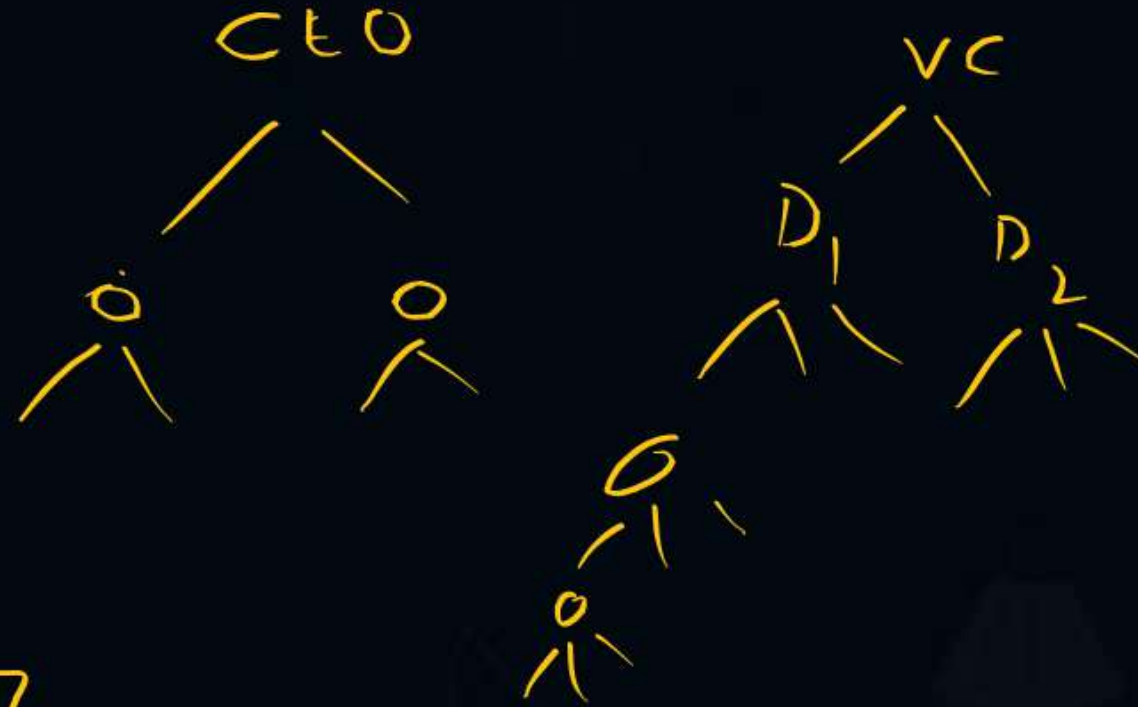
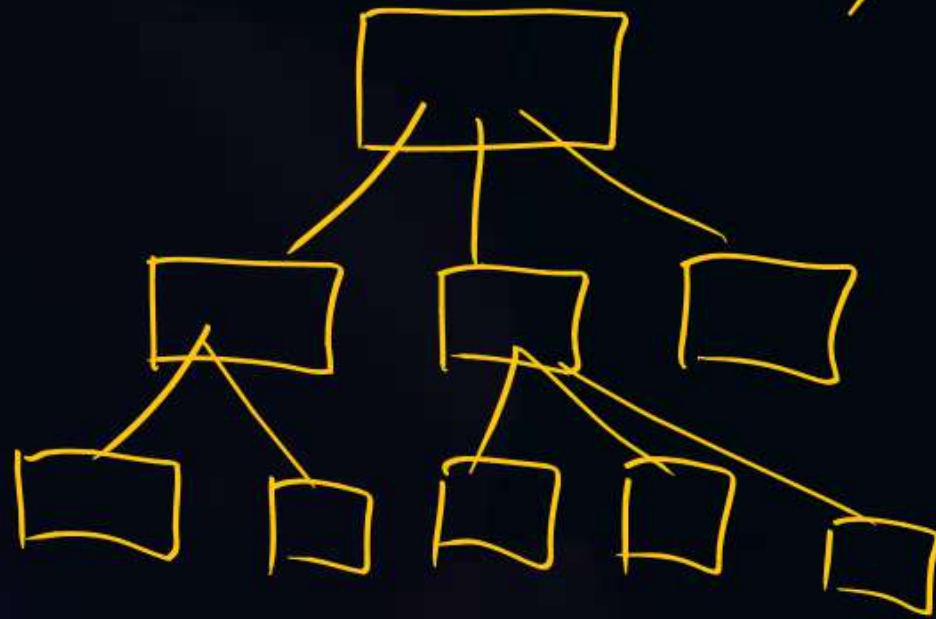


Topic : Tree



Tree

- 1) Structure of org
- 2) Folder



3.) HTML/XML (JSON objects)

* Non-linear data structure

4.) Binary search tree

5.) B-Tree, B⁺-Tree

6.) Parse tree

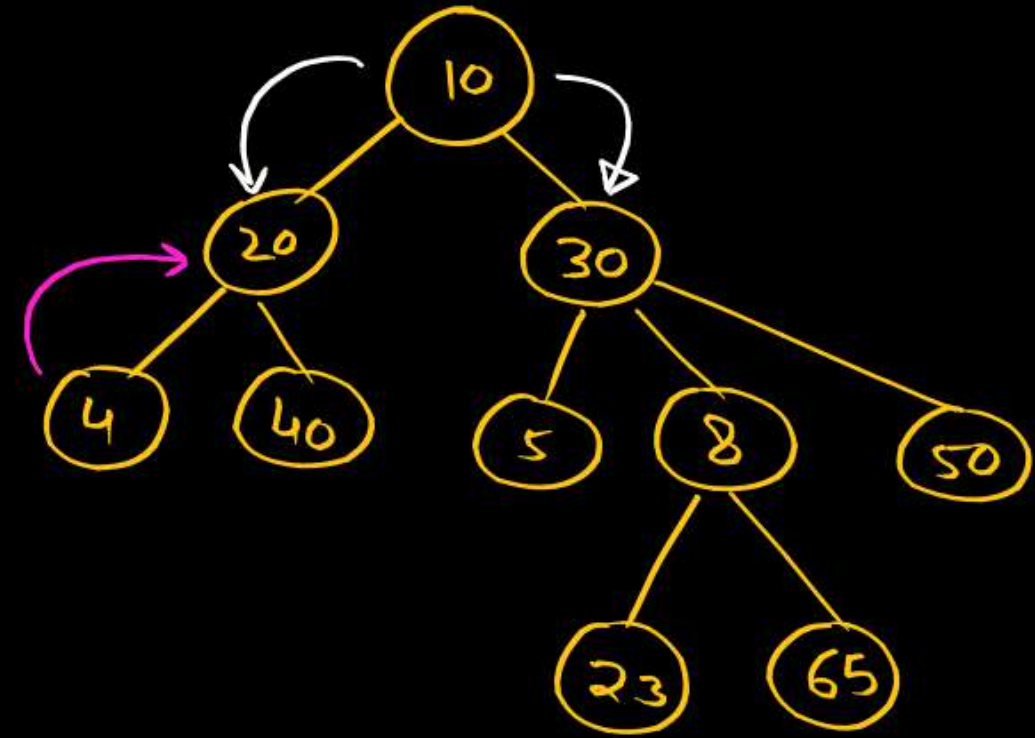
7.) Binary heap

1) Node : Element in a tree

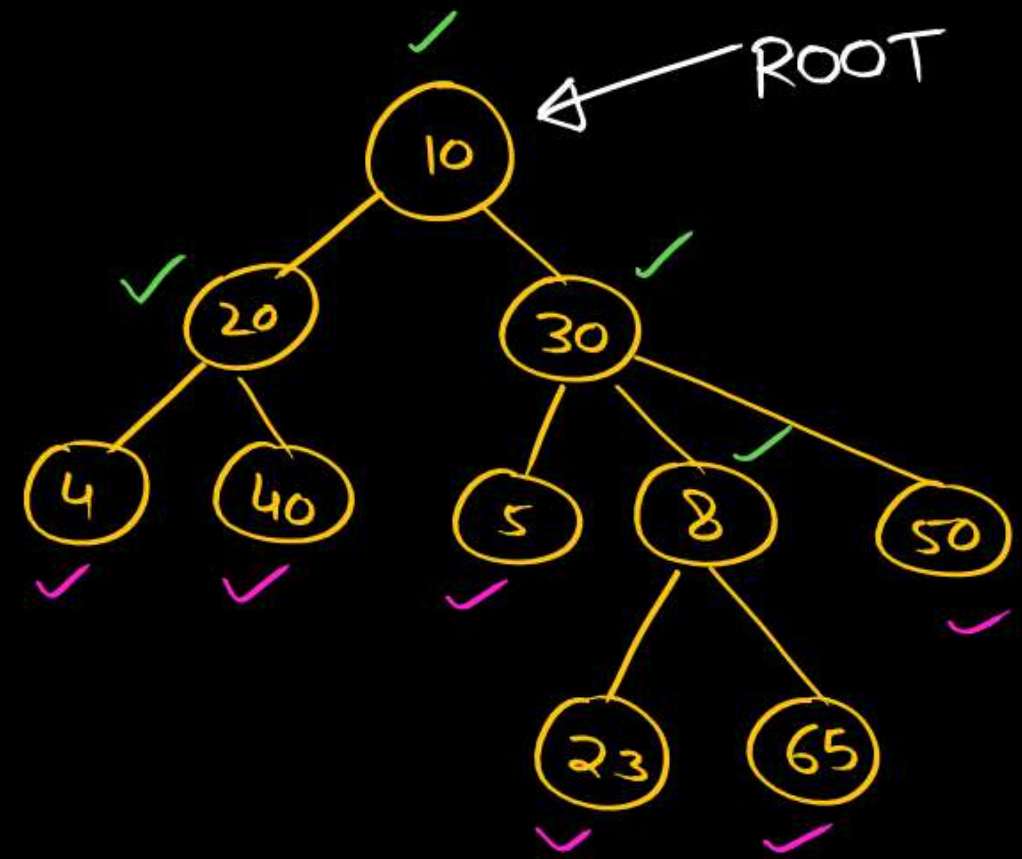
2) child : 20, 30 are children of node
with key 10

5, 8, 50 are children of node with
key 30.

3) Parent : 20 is the parent of 4
30 is the parent of 50
30 is the parent of 5, 8, 50



4.) ROOT : Only node without parent.
Distinguishable from all other nodes.



5.) Leaf node : A node with no child.
(External Node) 4, 40, 5, 23, 65, 50

6.) Internal Node : A node with atleast 1 child.
10, 20, 30, 8

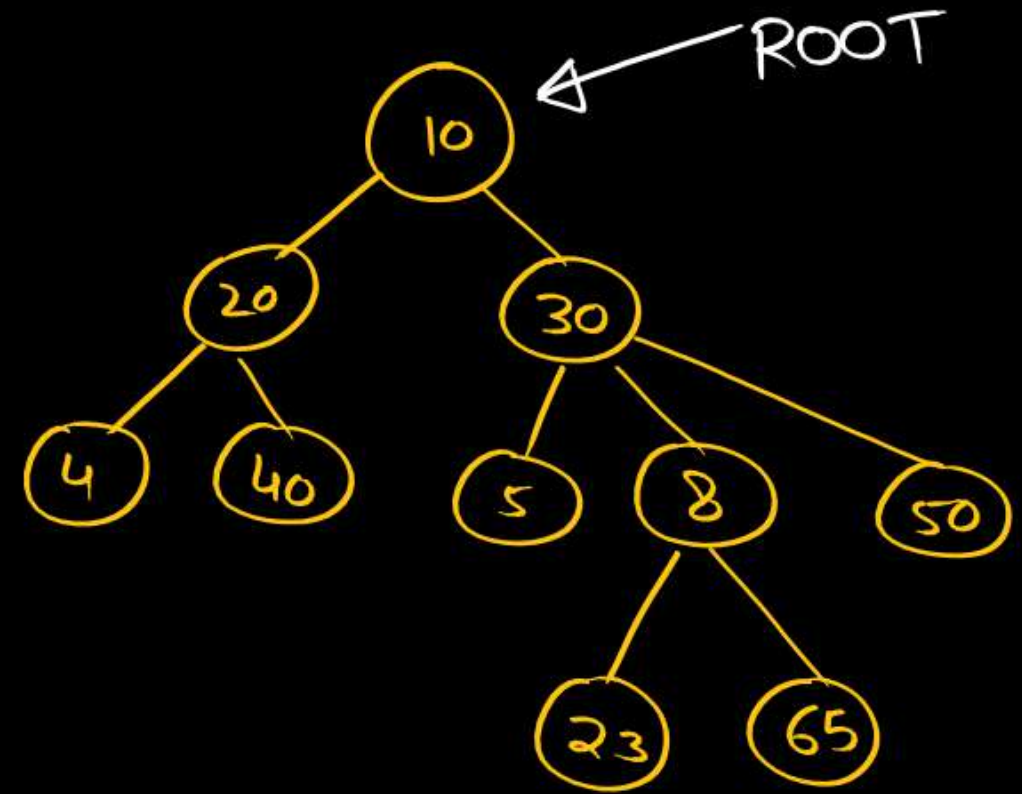
7.) Degree of a node :

no. of childs

degree of root node = 2

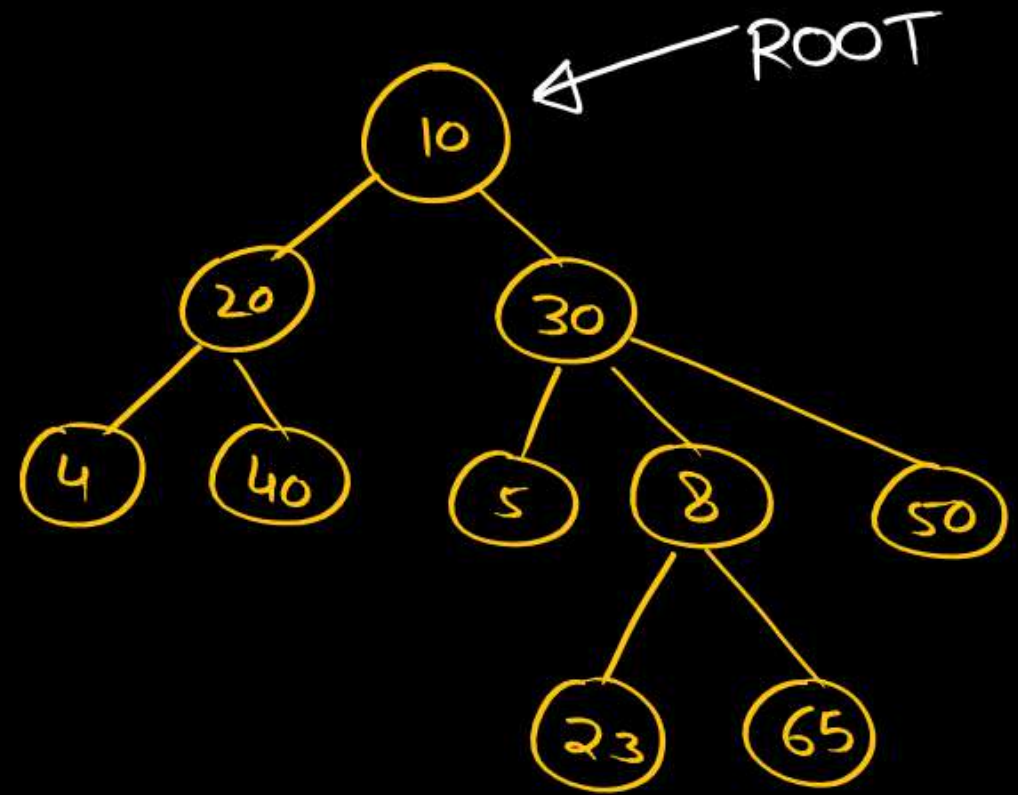
degree of node with key 30 = 3

degree of leaf node = 0

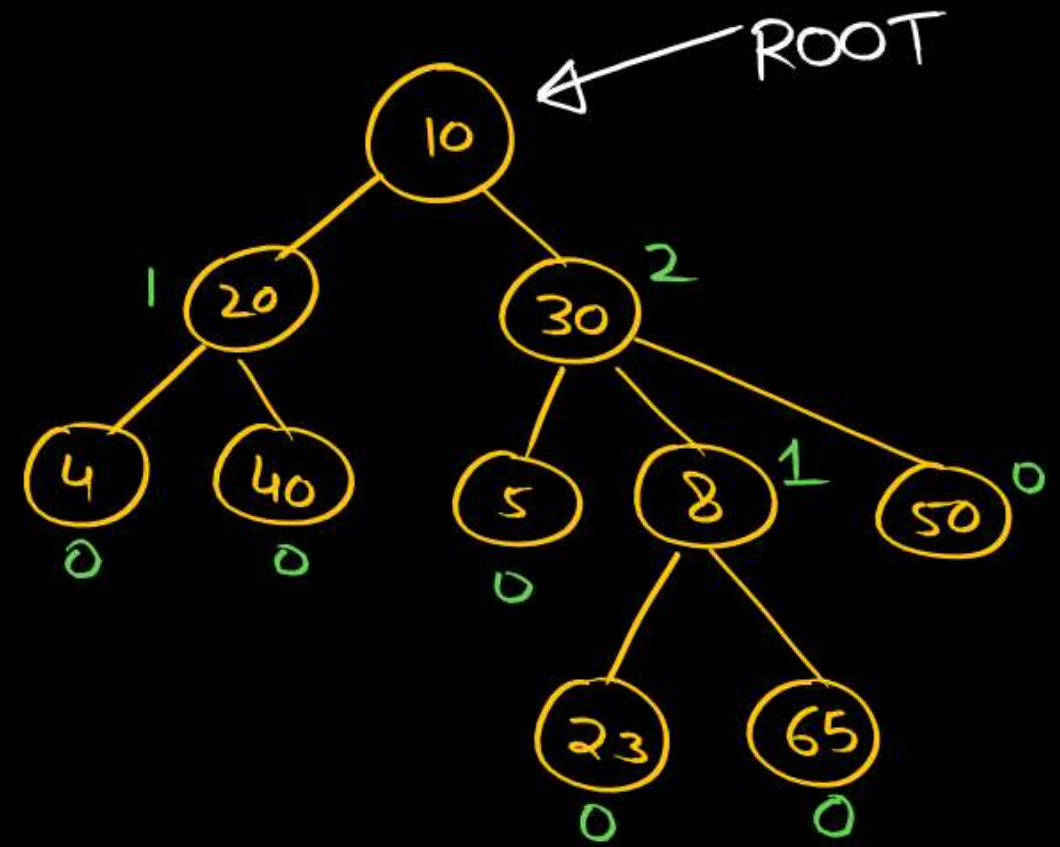


8.] height of a node : height of a node X is the length of path from node X to the farthest leaf node.

OR \Rightarrow the length of longest path from node X to any leaf node.



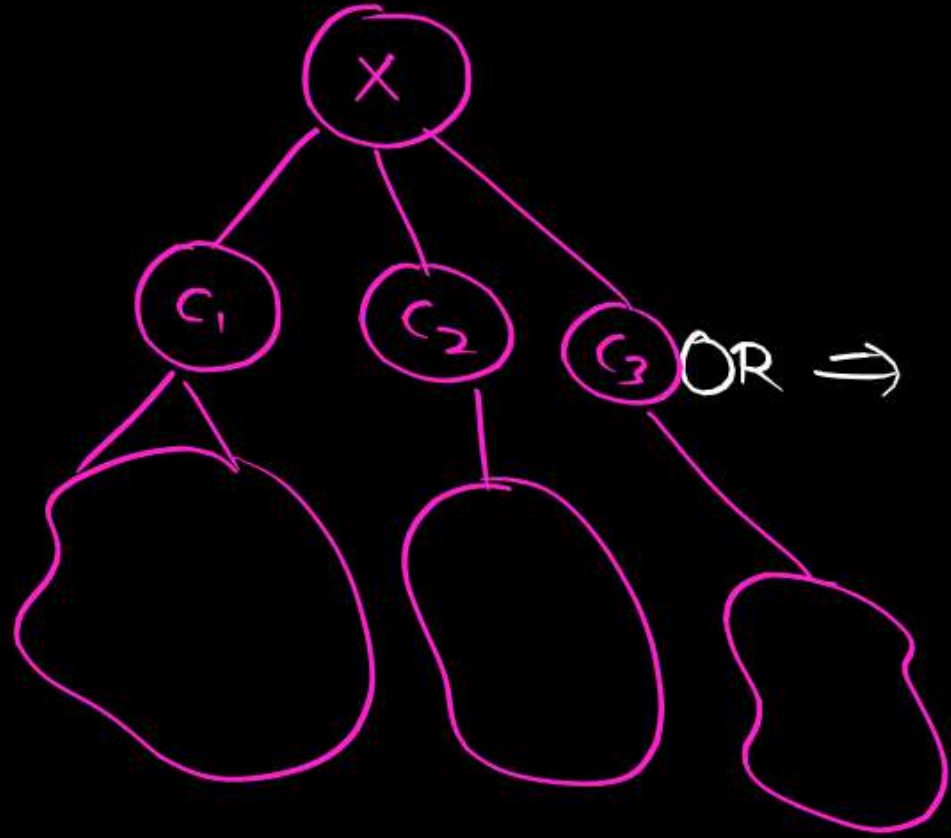
8.] height of a node : height of a node X is the length of path from node X to the farthest leaf node.



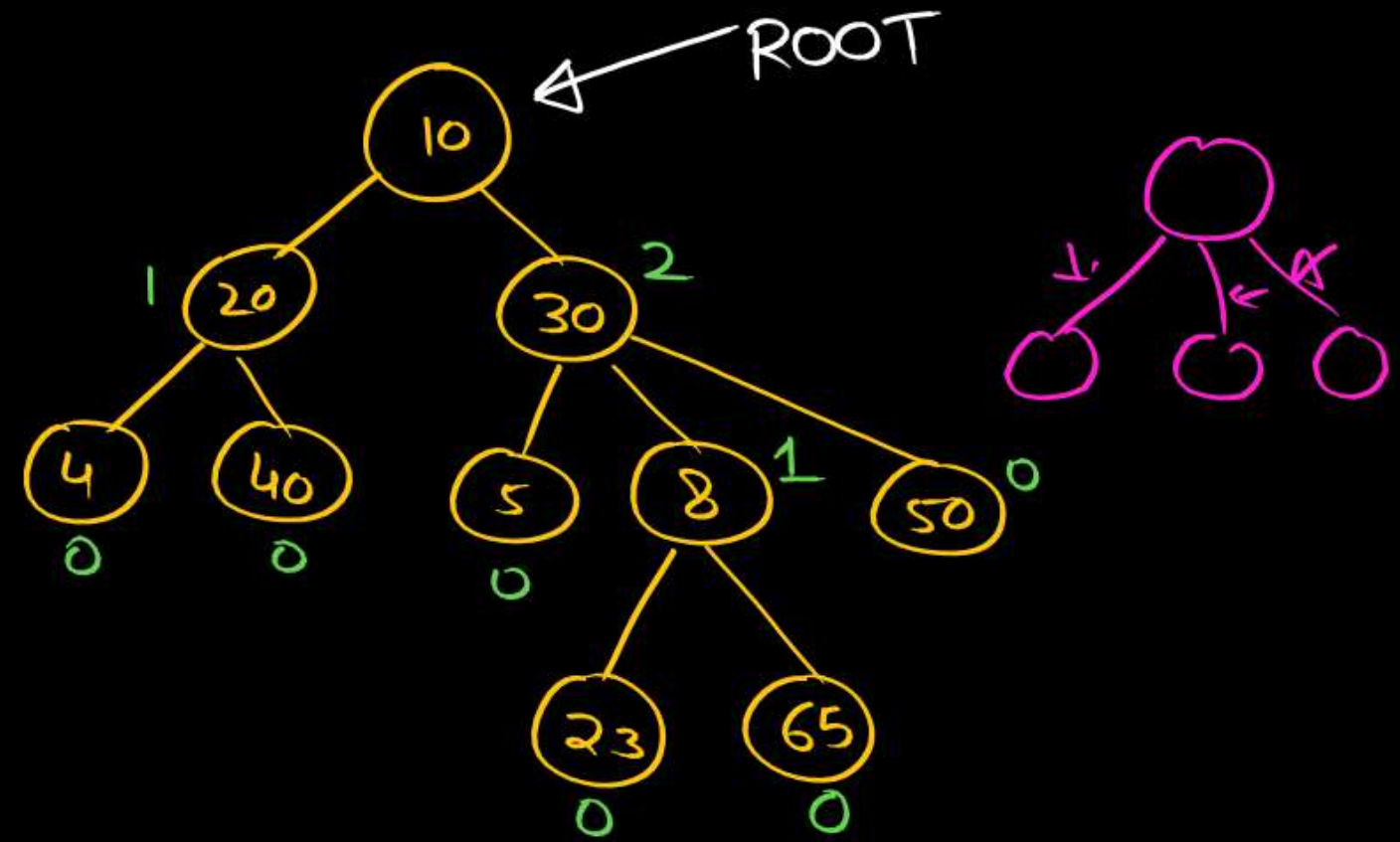
OR \Rightarrow the length of longest path from node X to any leaf node.

① height of leaf node = 0

8.] height of a node : height of a node X is the length of path from node X to the farthest leaf node.



the length of longest path from node X to any leaf node.

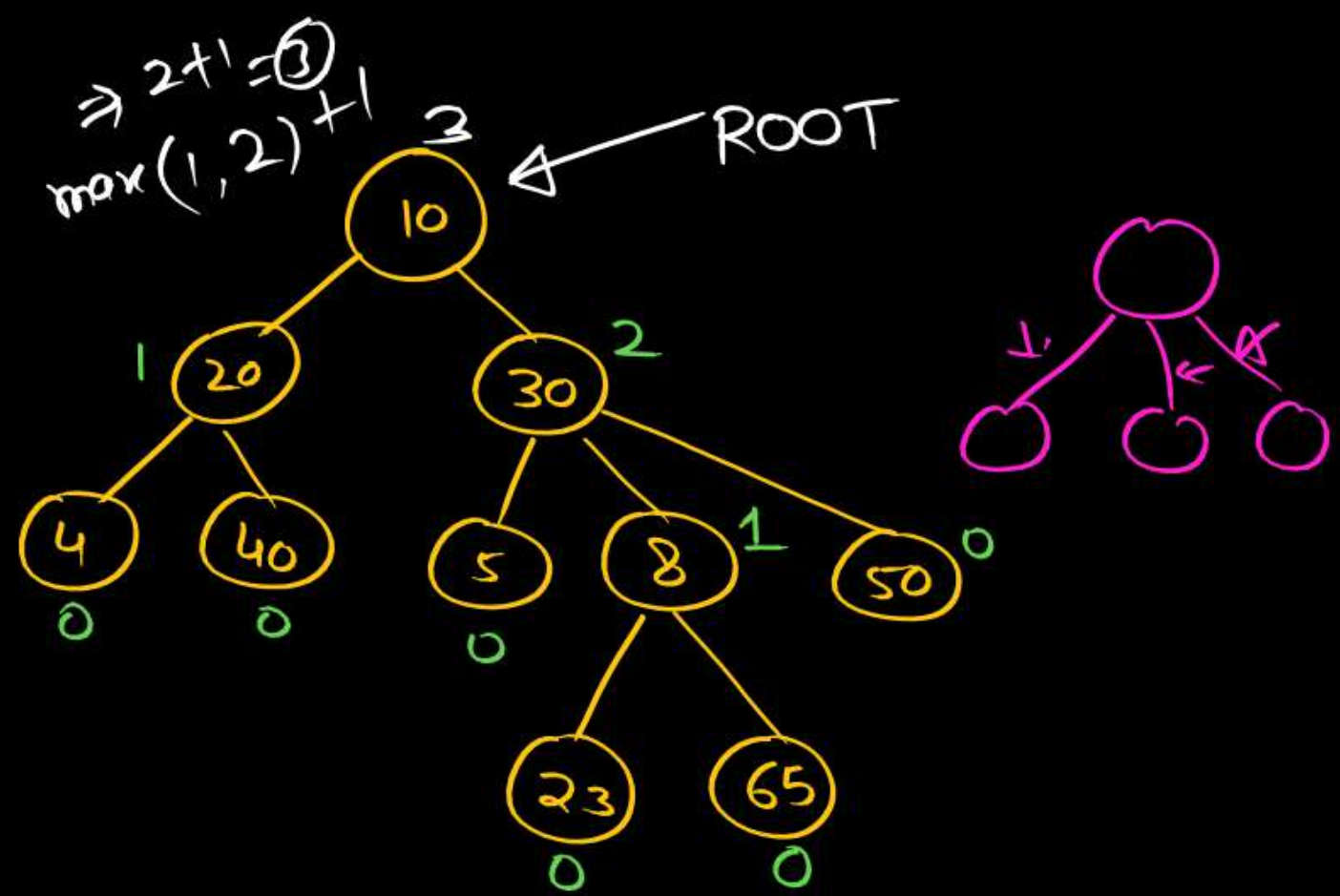


① height of leaf node = 0

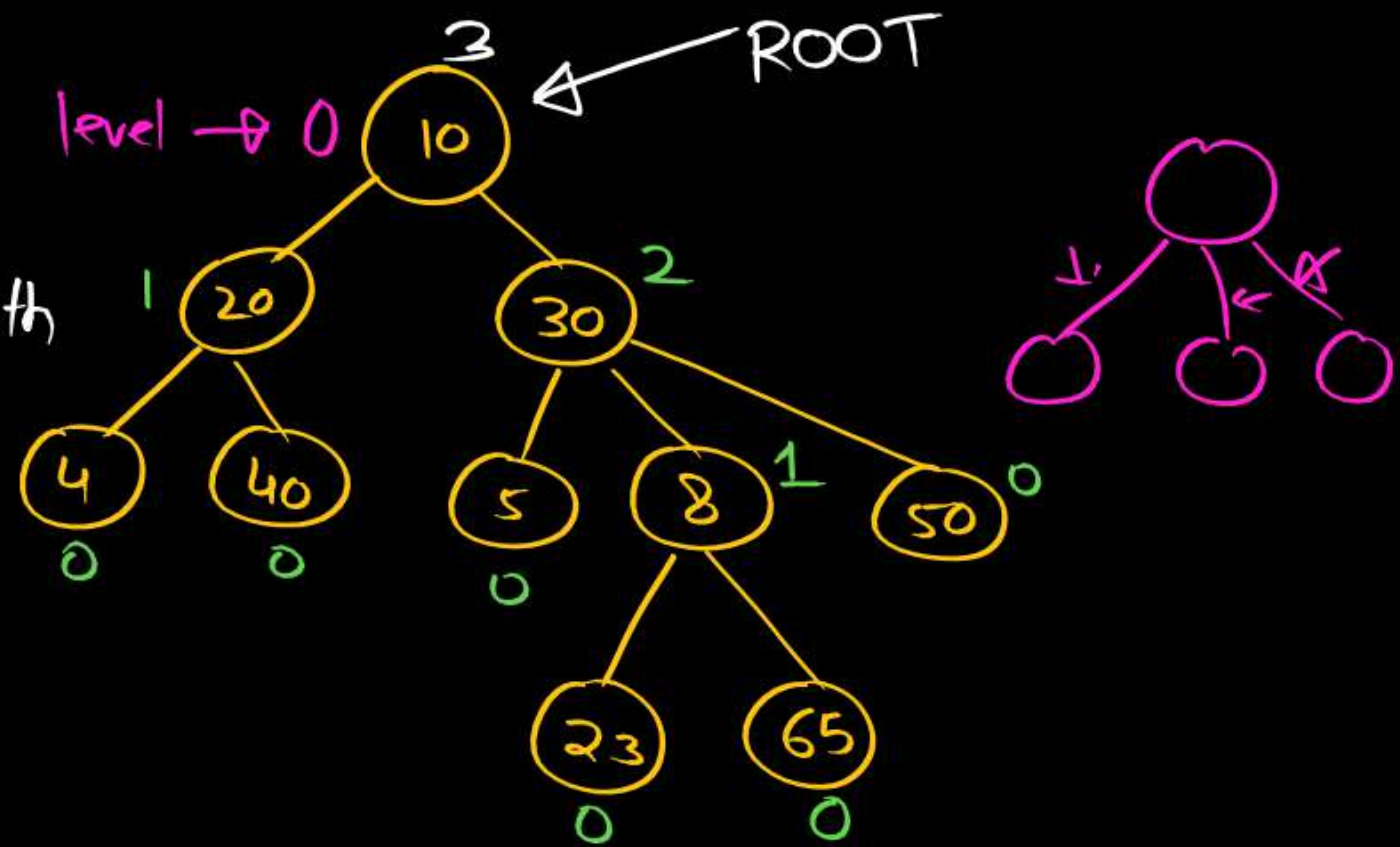
$$h(\text{Node}) = \max(\text{height of all childs}) + 1$$

q.) Height of a tree

= height of root node



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

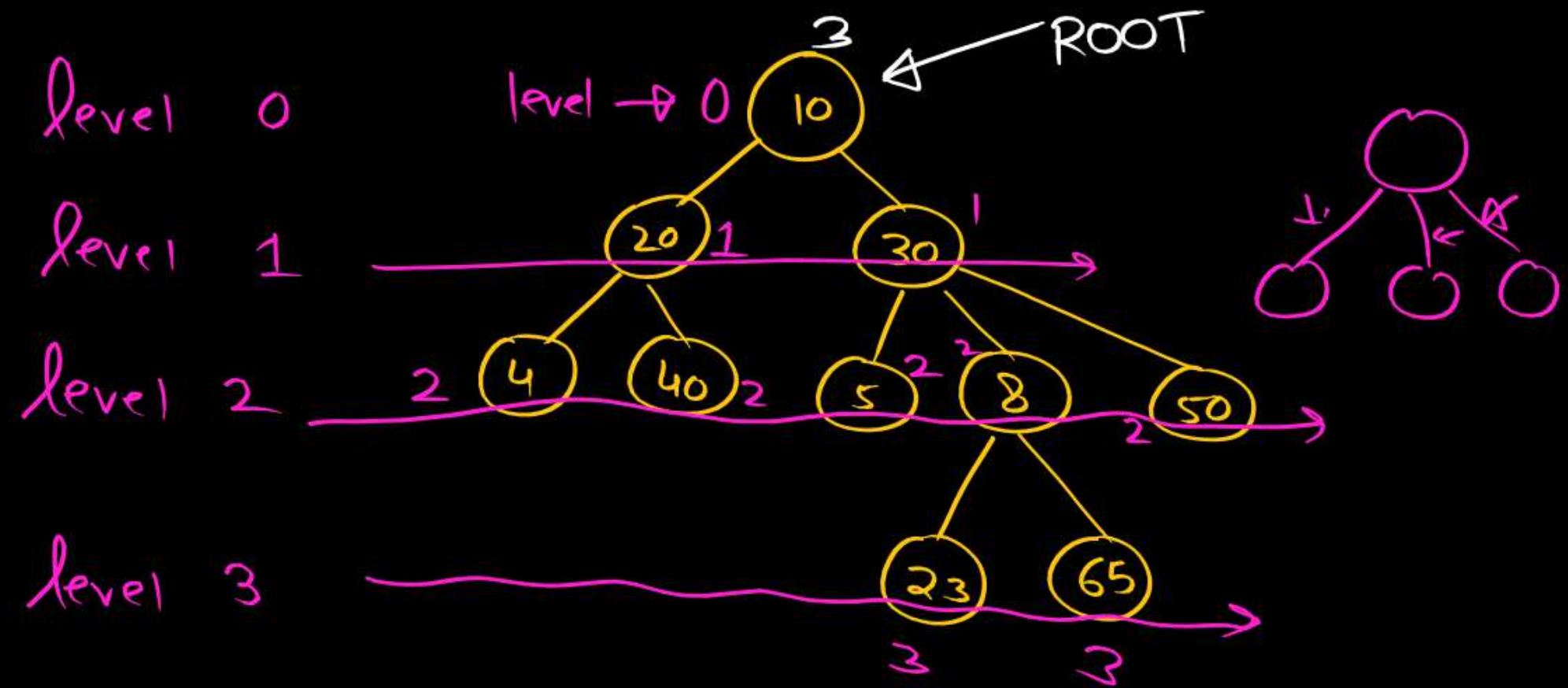


level of root node = 0

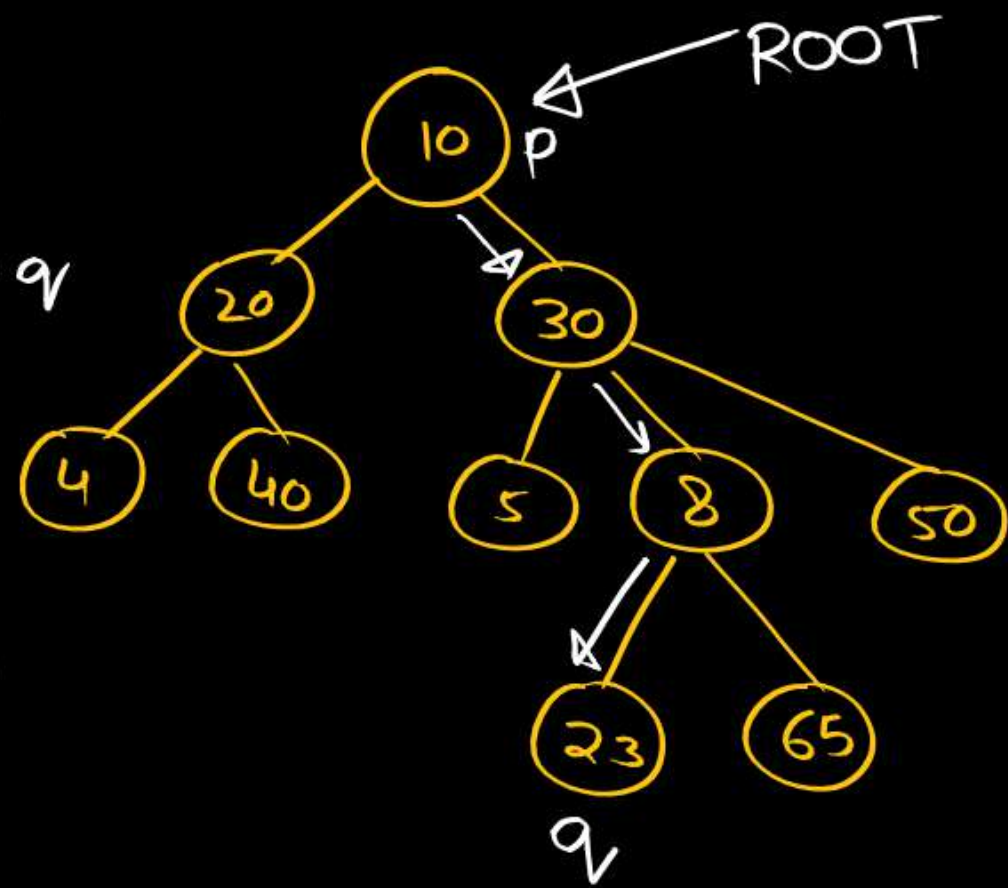
level of any node = (level of parent node) + 1

10.

level/depth of a node

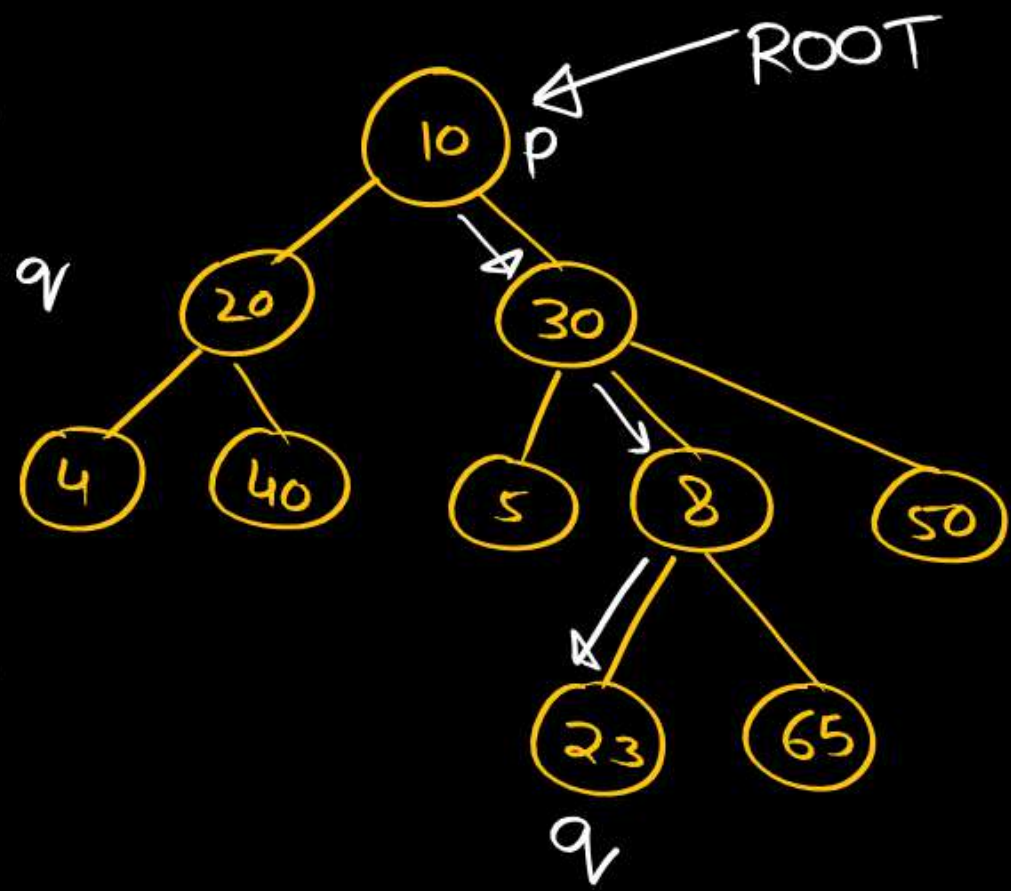


11. Ancestor of a node : If there is a path
from node P to node q
then all the nodes
in the path (other than q)
are ancestor of node q .



8 is the ancestor of 23
30 is the ancestor of 23
10 is the ancestor of 23

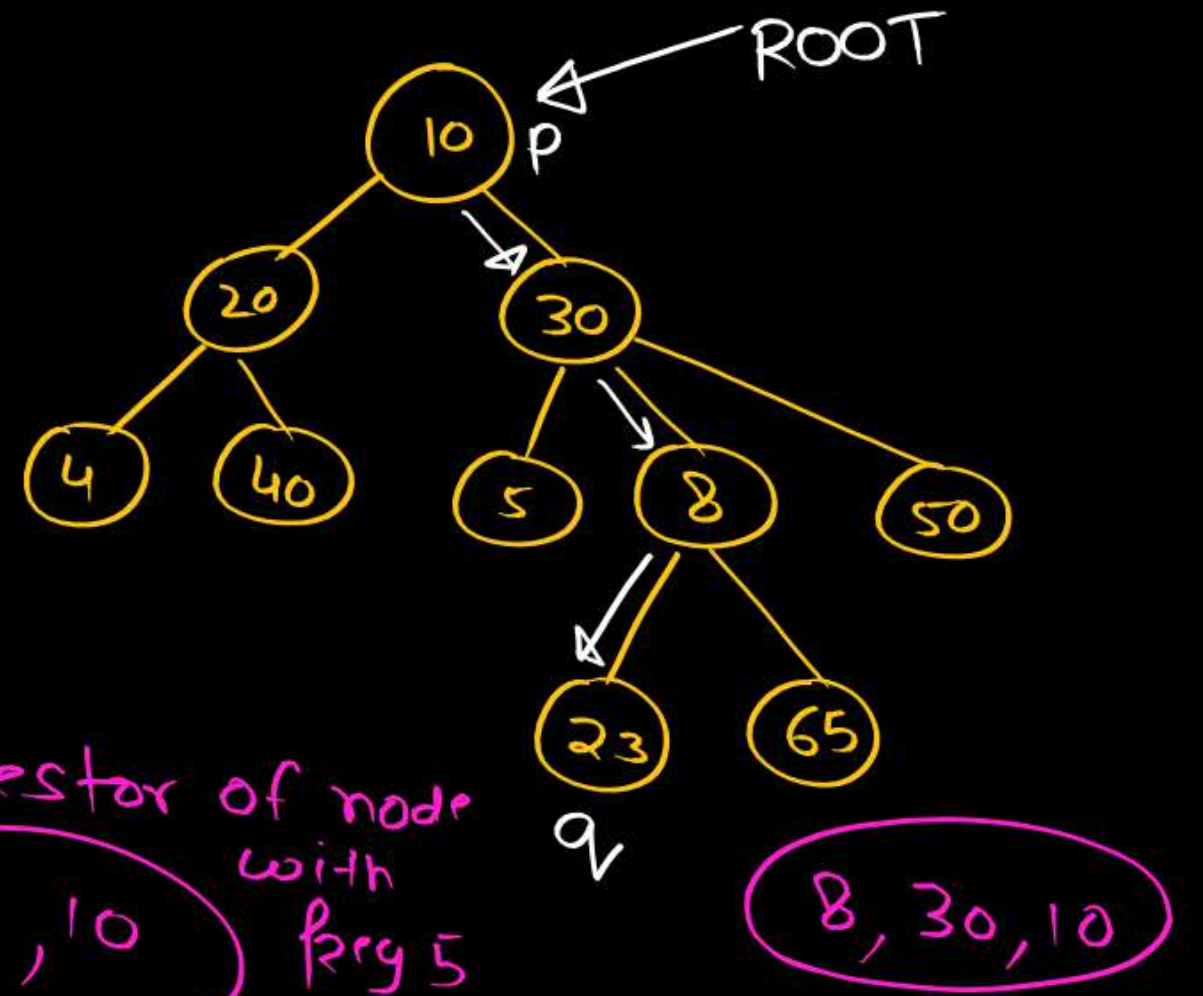
11. Ancestor of a node : If there is a path from node P to node q then all the nodes in the path (other than q) are ancestor of node q .
OR



- (i) Parent of a node is an ancestor of node
- (ii) Parent of some ancestor is also an ancestor.

8 is the ancestor of 23
30 is the ancestor of 23
10 is the ancestor of 23

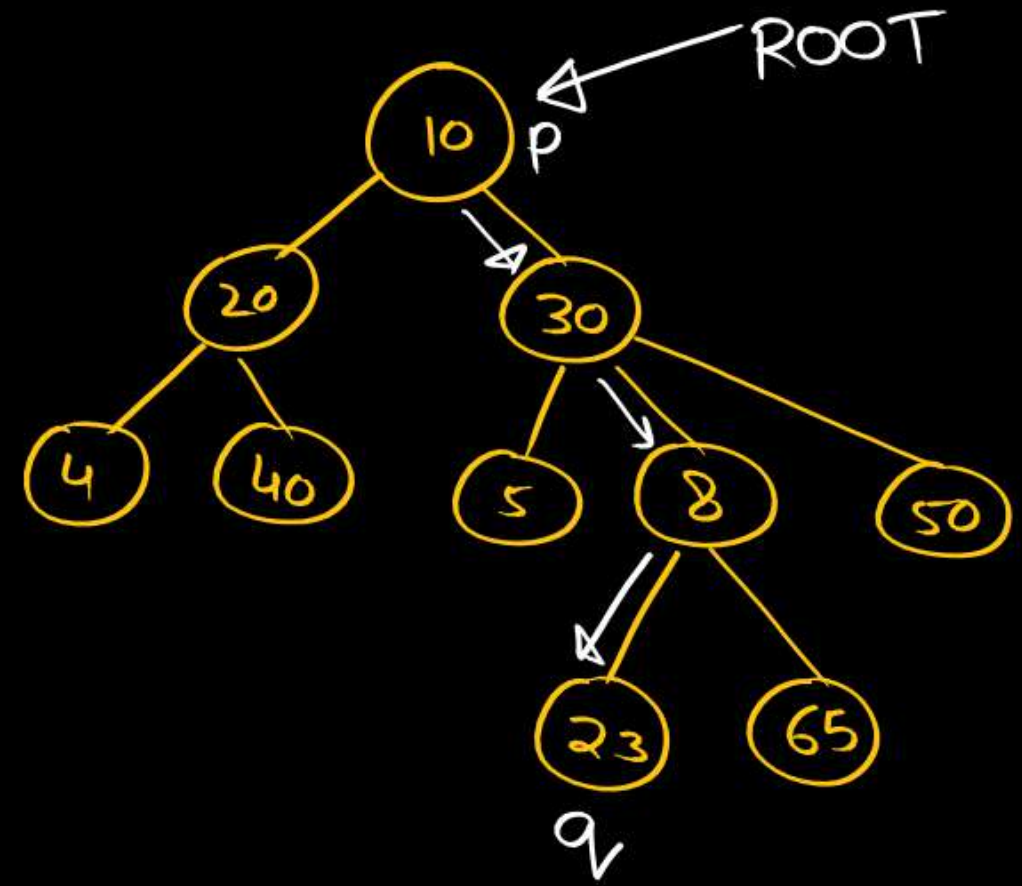
10 is the ancestor
of
all the nodes
(except itself)



12. descendant :

If p is an ancestor of q

$\Rightarrow q$ is a descend. of p .

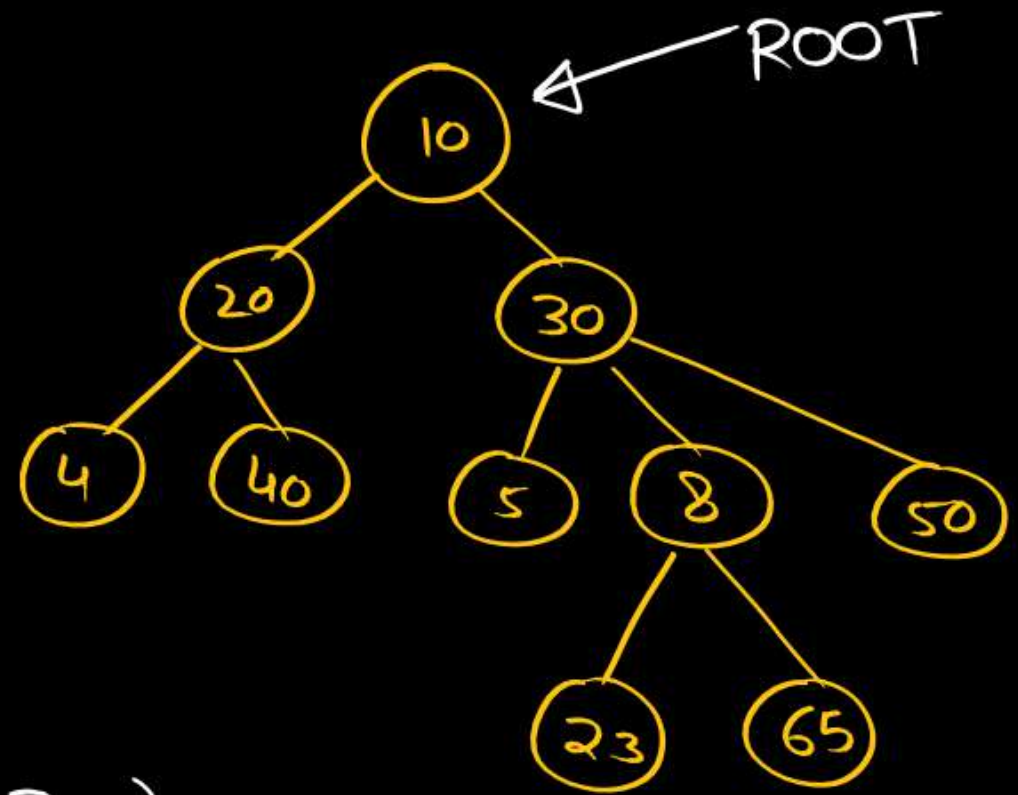


13.) Siblings : Nodes having same parent.

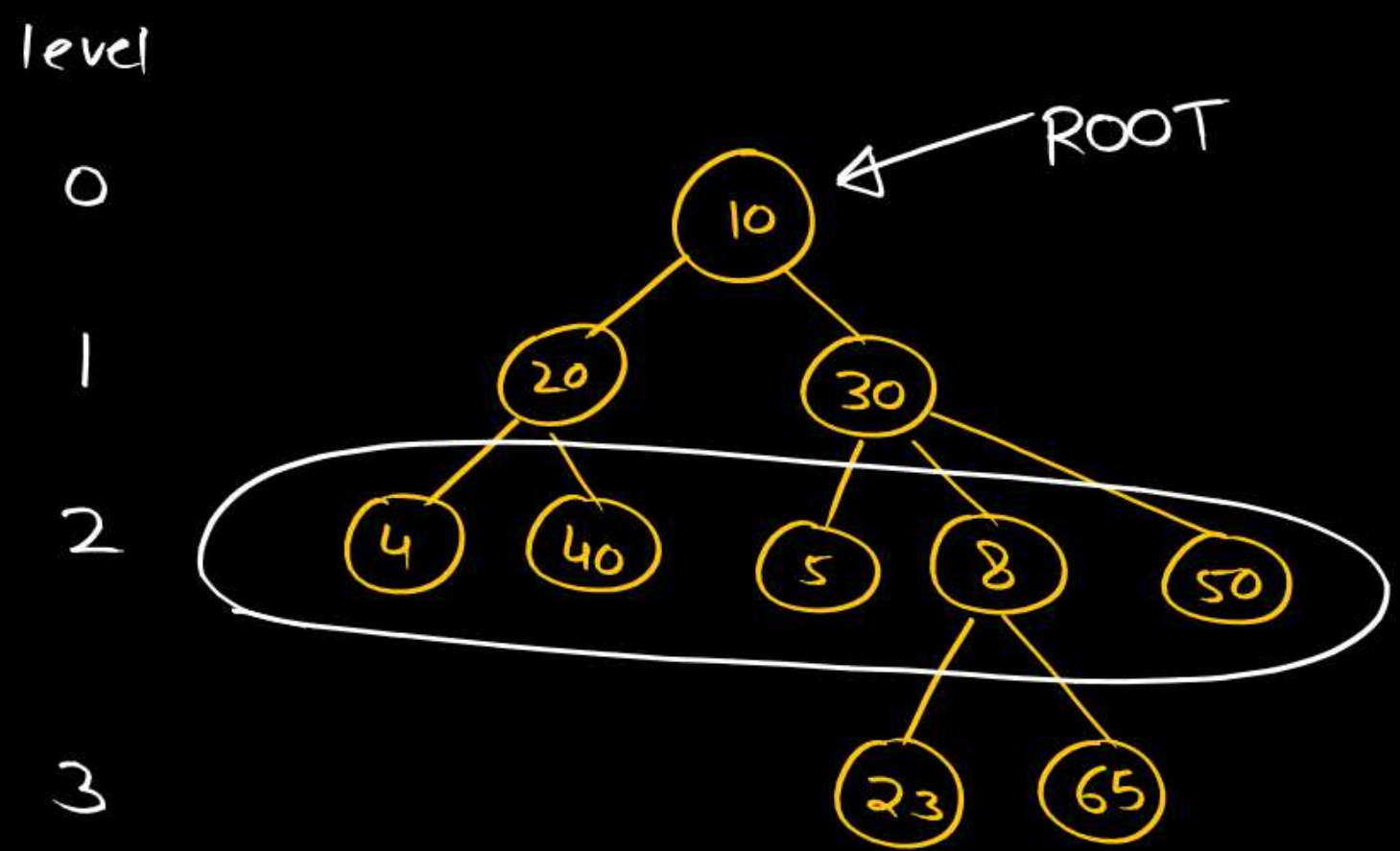
20, 30 \Rightarrow Siblings (Parent 10)

40, 5 \Rightarrow Not Siblings

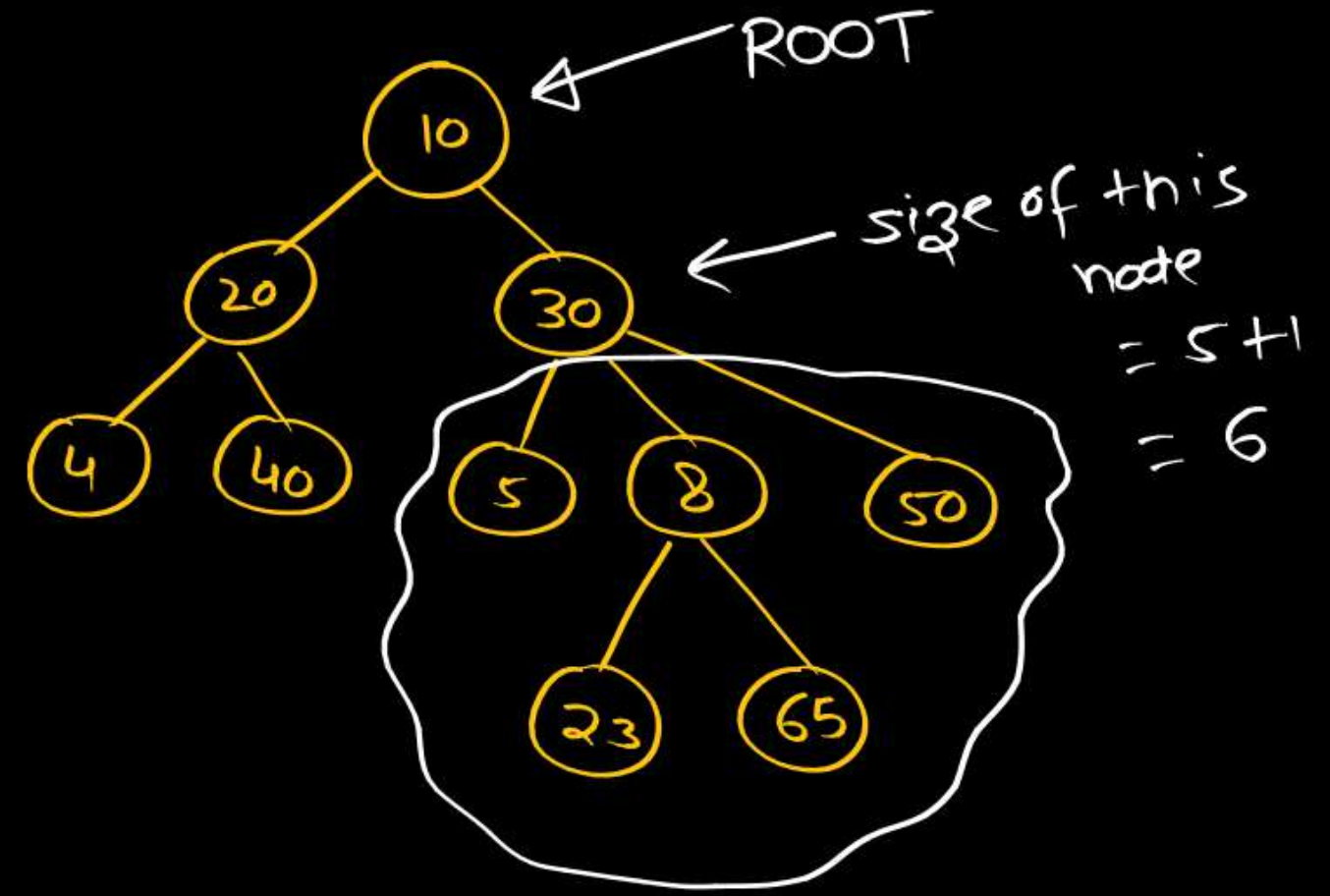
5, 8, 50 \Rightarrow Siblings (Parent 30)



14. Generation: Nodes at any particular level form a generation.

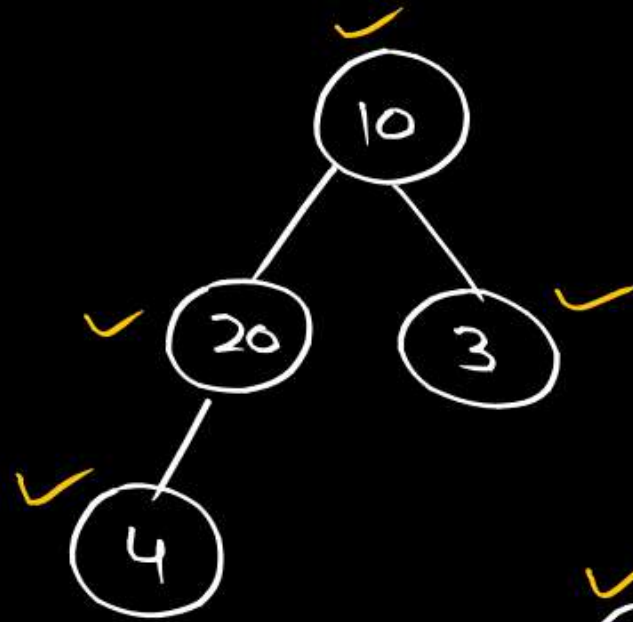


15. Size of a node : The no. of
descendant of
a node (including
itself)

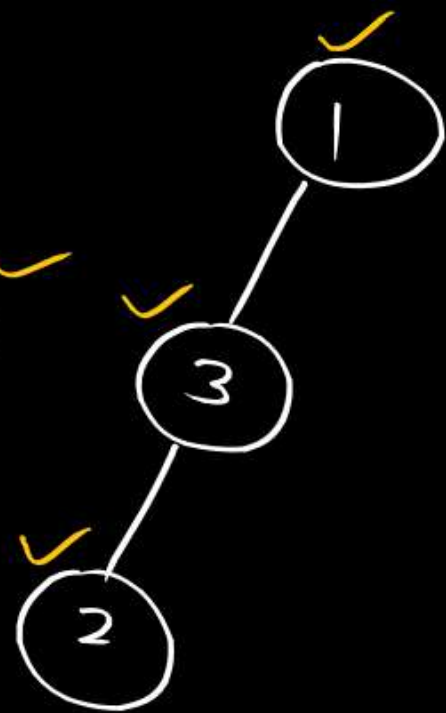


Binary Tree

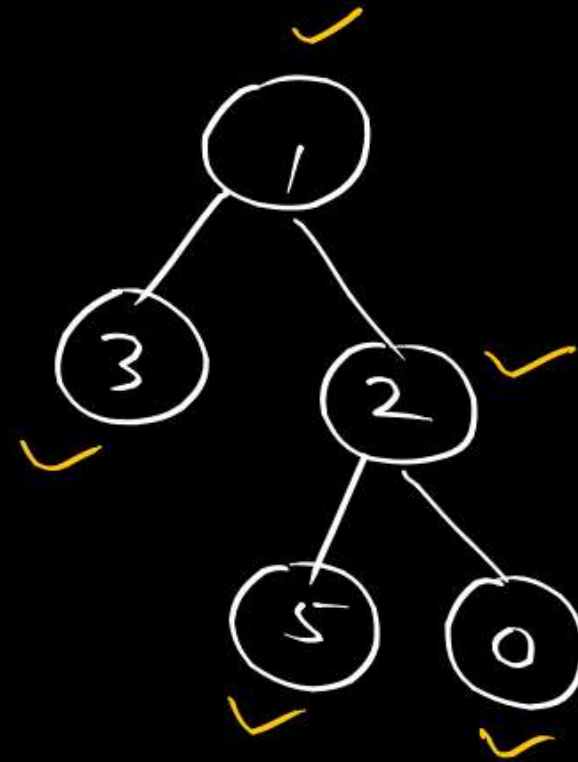
A node can have at most 2 childs.



Binary tree

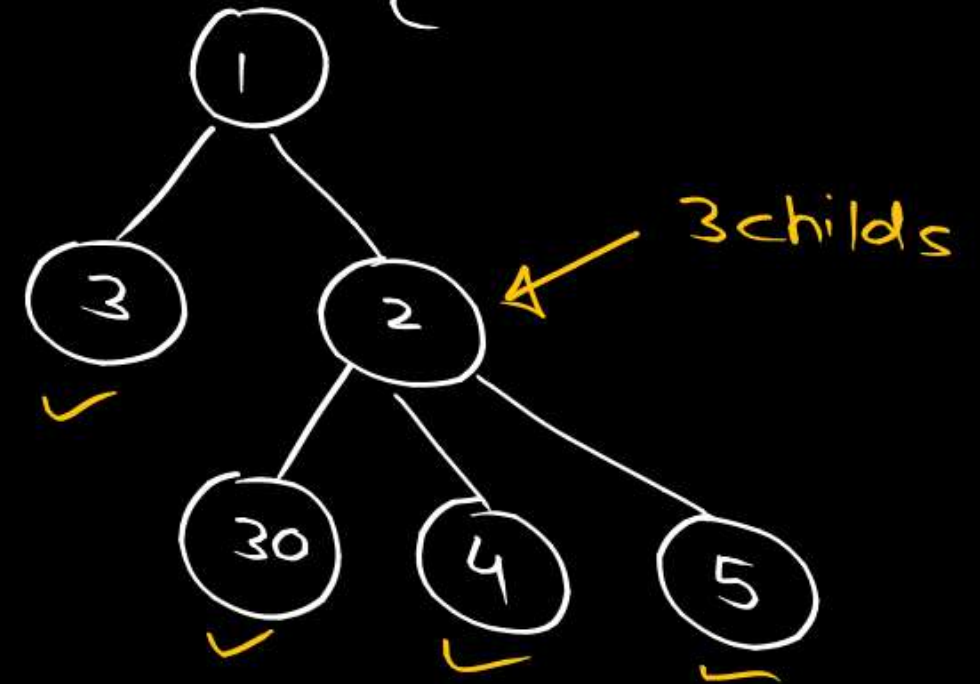


Binary tree

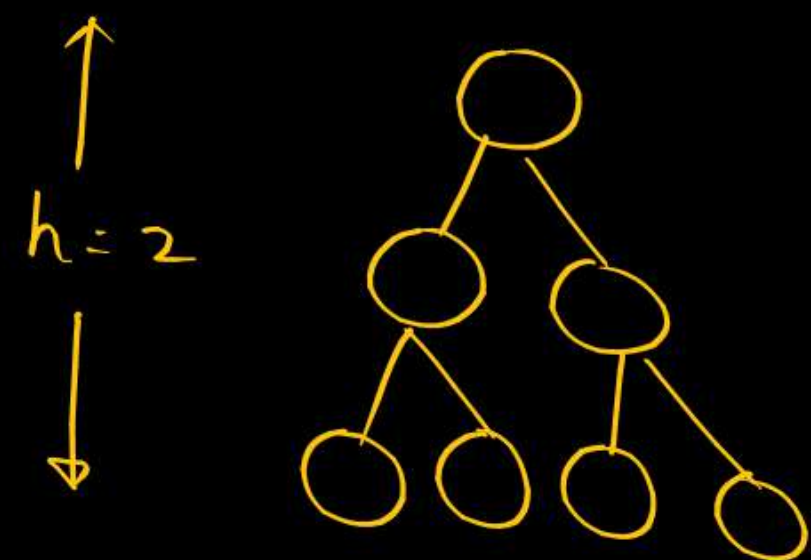
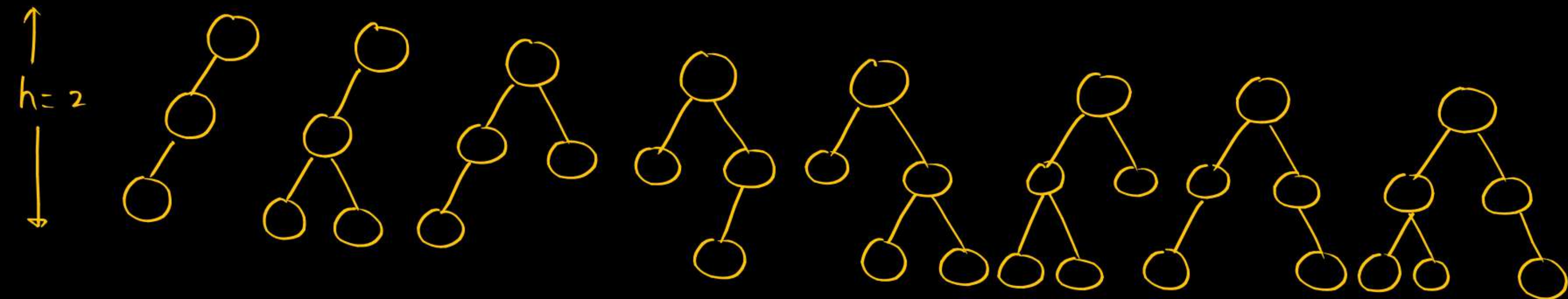


Binary tree

leaf — 0 child
Internal node { — 1 child
 — 2 child

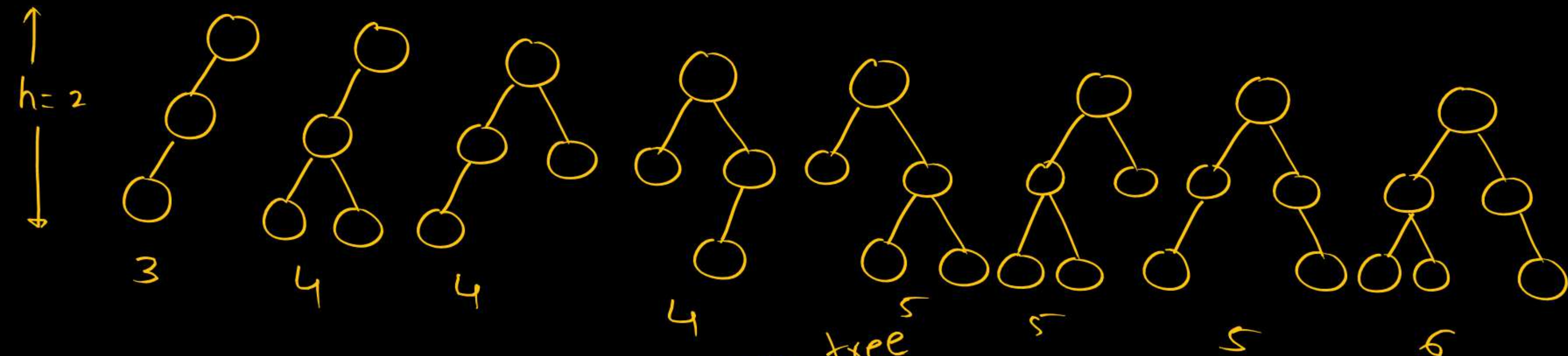


Binary tree X

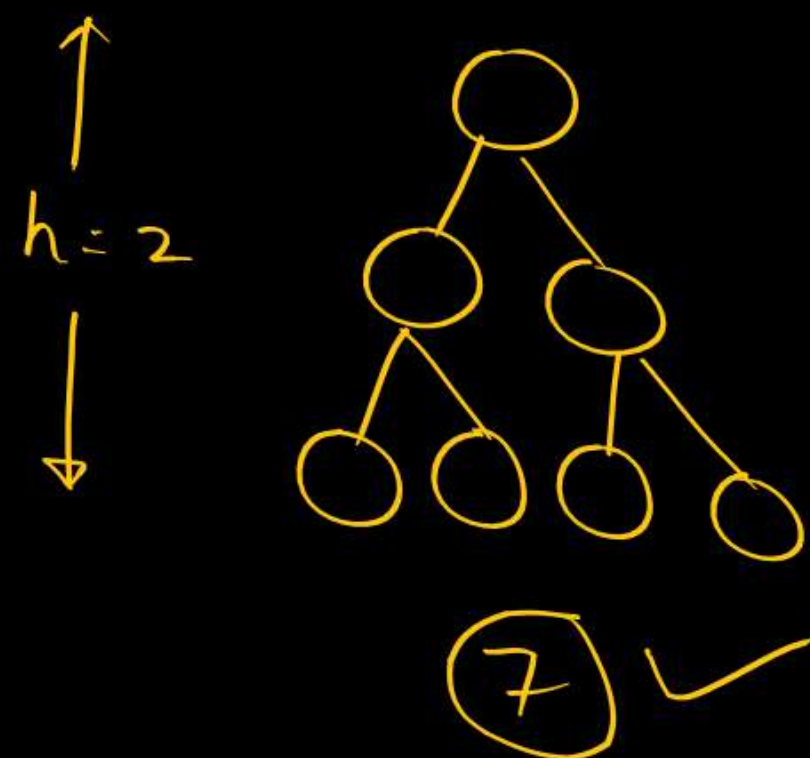


→ Full binary tree

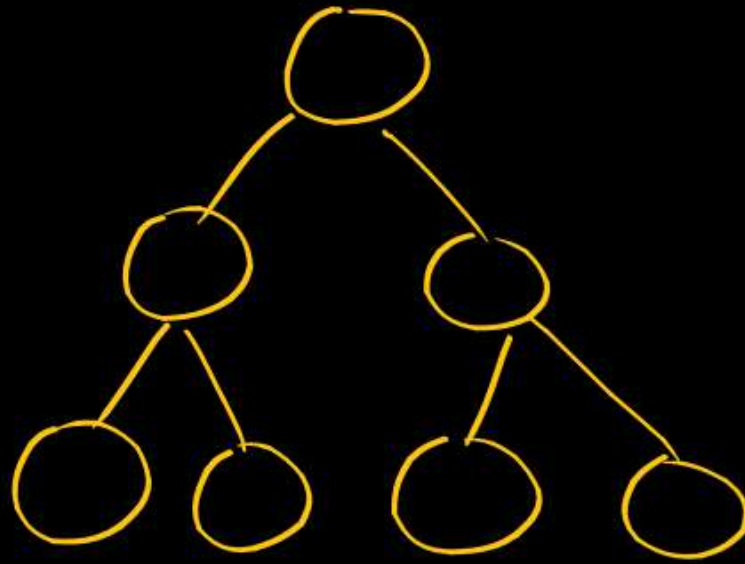
What is the maximum no. of nodes in a binary tree of height h ?



→ Full binary tree



$h=2$



level # of nodes

0

1

1

2^1

2

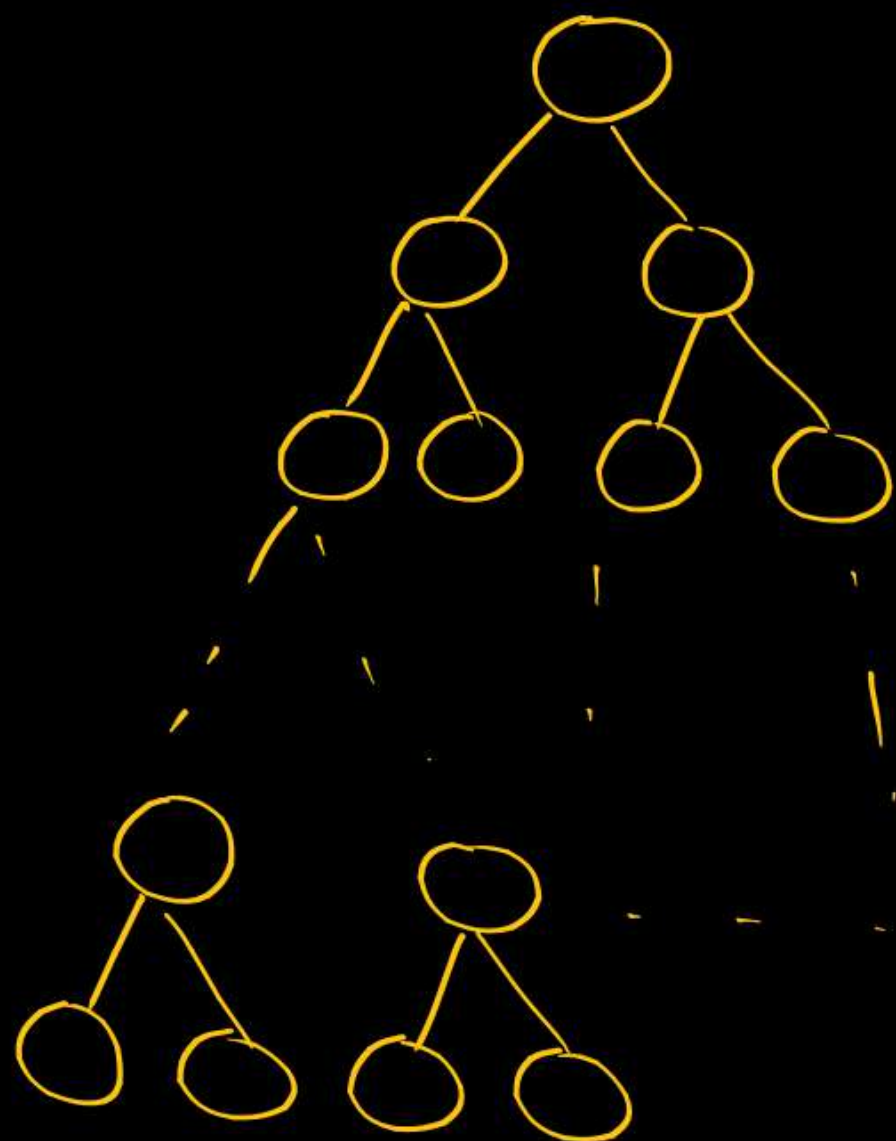
2^2

$$\# \text{ nodes} = 1 + 2^1 + 2^2 \quad (3 \text{ term})$$

$$= \frac{1(2^3 - 1)}{2 - 1}$$

$$= \frac{2^3 - 1}{1} = 2^3 - 1 = 7$$

$$\frac{a(r^n - 1)}{r - 1}$$



level

nodes

0

1

1

2^1

2

2^2

\vdots

\vdots

$h-1$

2^{h-1}

h

2^h

$$n = h+1$$

$$a = 1$$

$$r = 2$$

$$S_n = \frac{a(r^n - 1)}{r - 1}$$

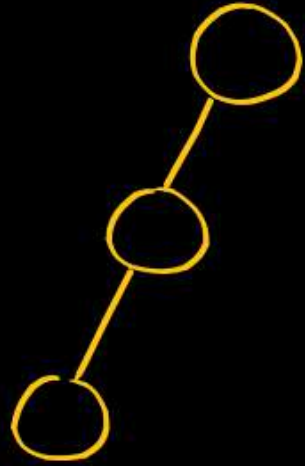
$$\# \text{ nodes} = \frac{2^{h+1} - 1}{2 - 1}$$

$$\# \text{ nodes} = 1 + 2^1 + 2^2 + \dots + 2^{h-1} + 2^h$$

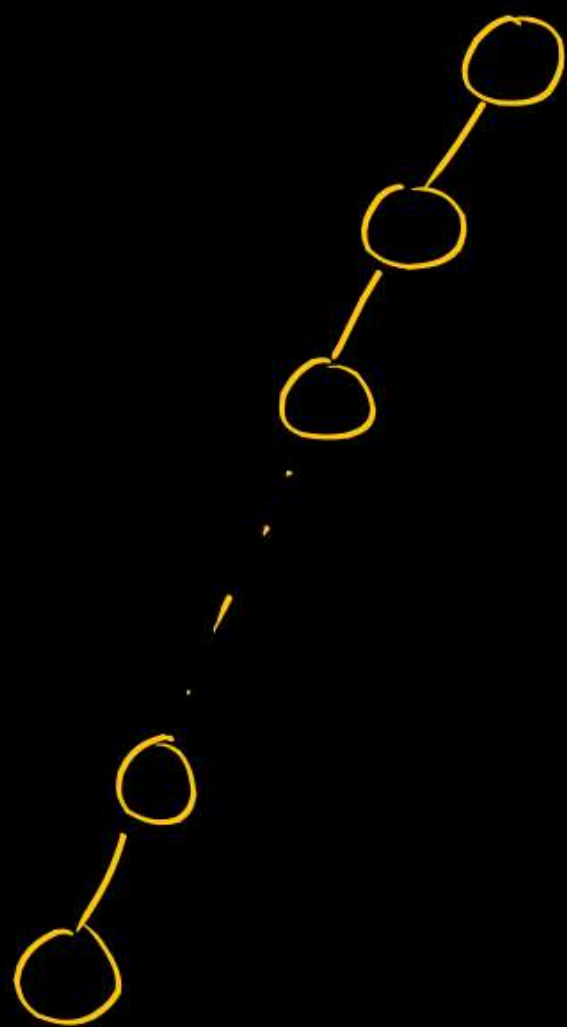
$$n_{\max} = 2^{h+1} - 1$$

What is the minimum no. of nodes in a binary tree of height h ?

$h=2$



$n_{\min} = 3 \quad (h=2)$



level # nodes

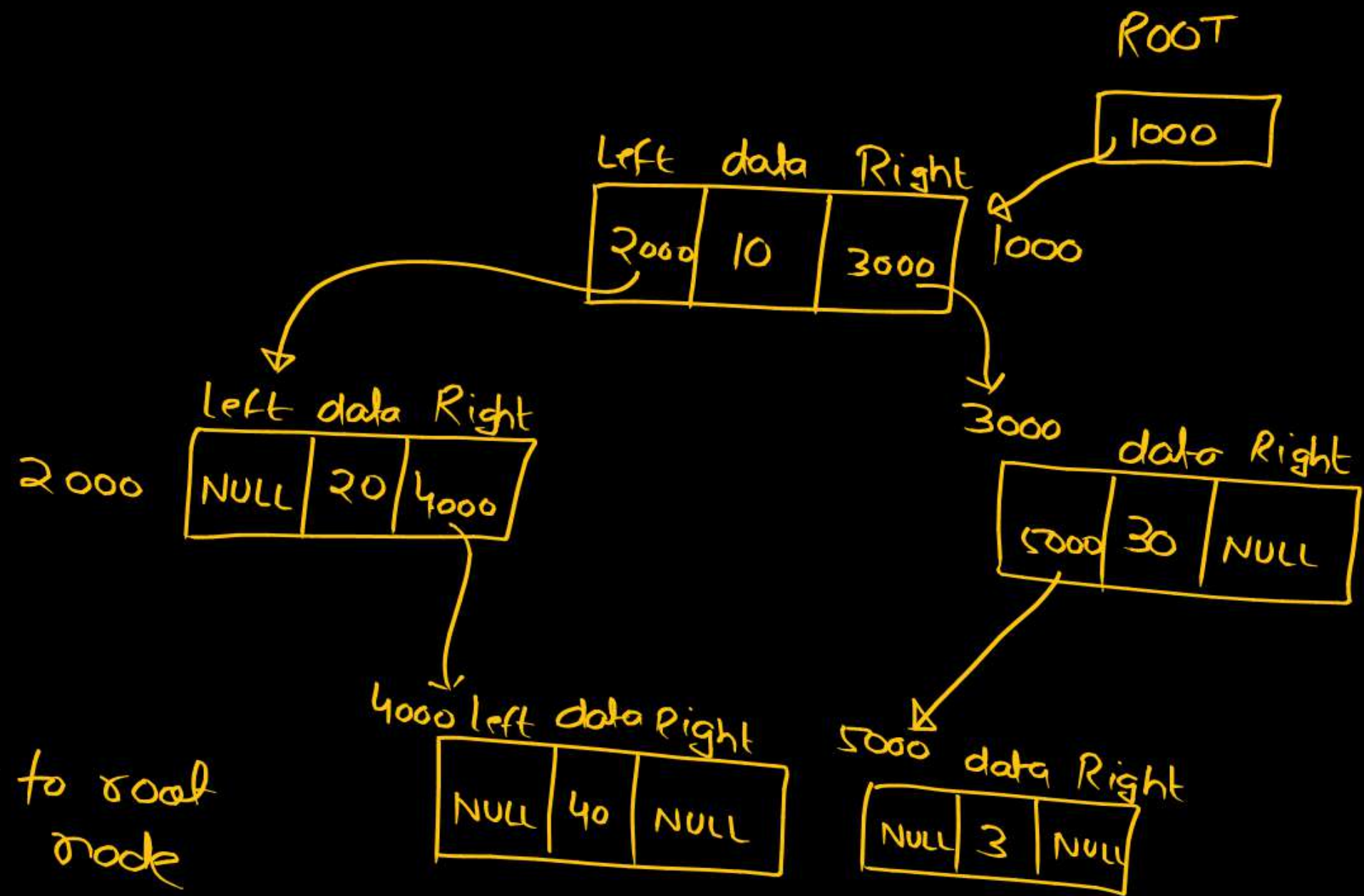
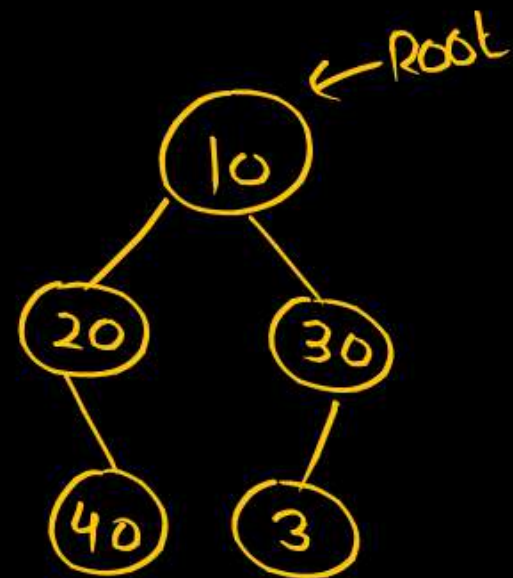
0	1
1	1
2	1
⋮	⋮
⋮	⋮
$h-1$	1
h	1

$$n_{\min} = 1 + 1 + 1 + \dots + 1 \quad (h+1 \text{ times})$$

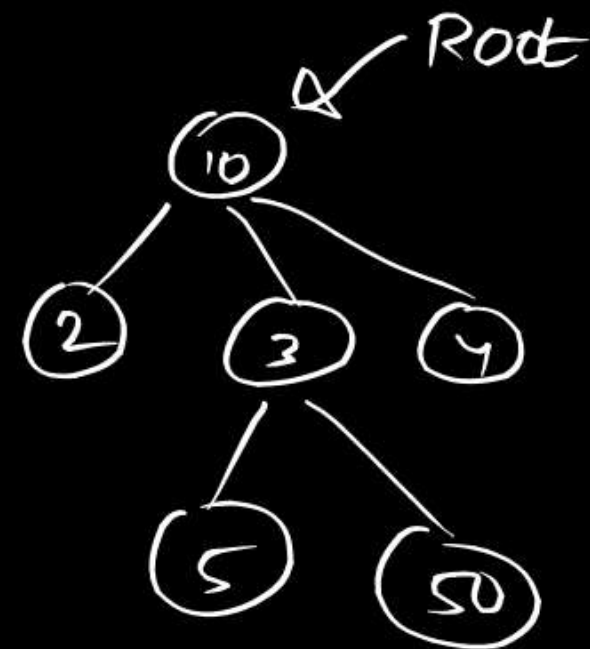
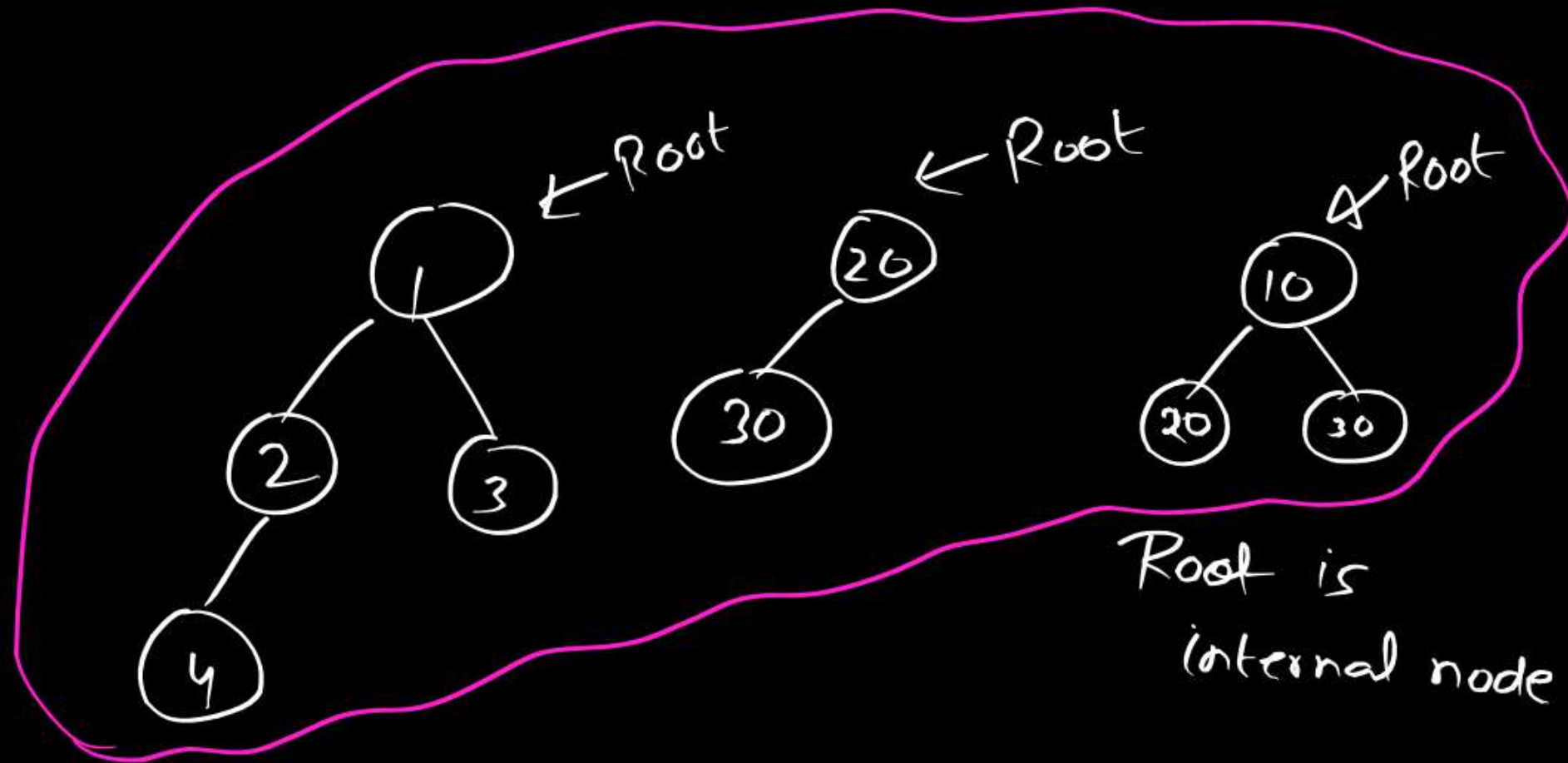
$$n_{\min} = h + 1$$

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

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



ROOT : Pointer to root node



THANK - YOU