unacademy

Trees - Part II

Course on Data Structure

Pankaj Sharma • Lesson 22 • Aug 1, 2024

unacademy

CS & IT Engineering

Data Structure

Tree

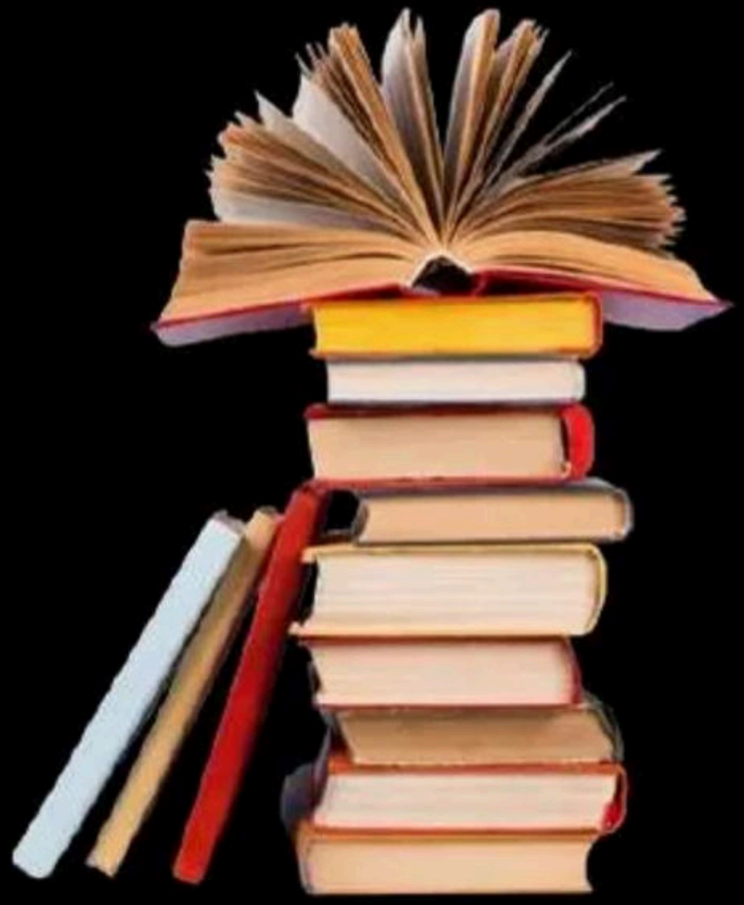
Lecture Number- 22

By- Pankaj Sir

unacademy

Topics

to be covered



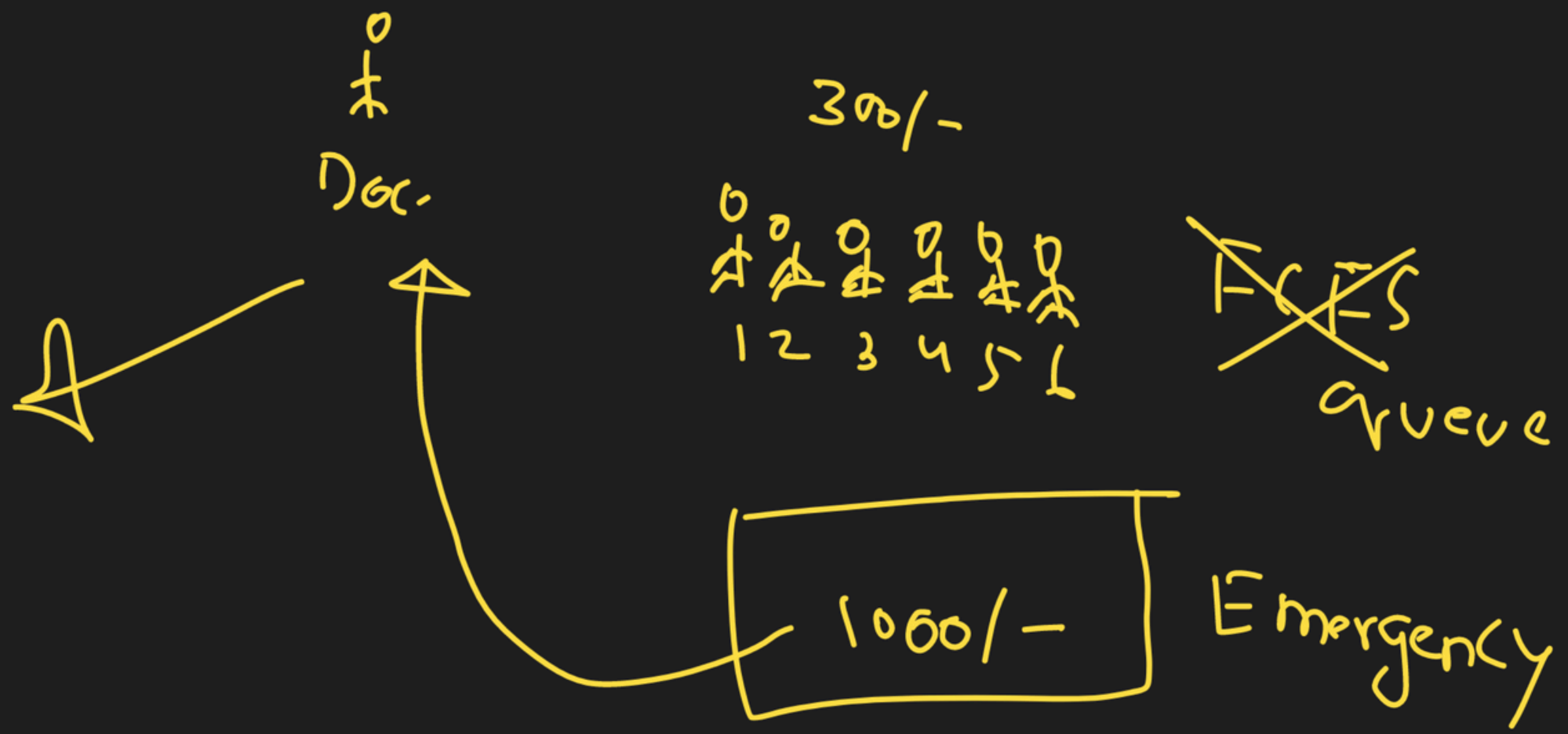
1. Tree-I

Priority queue

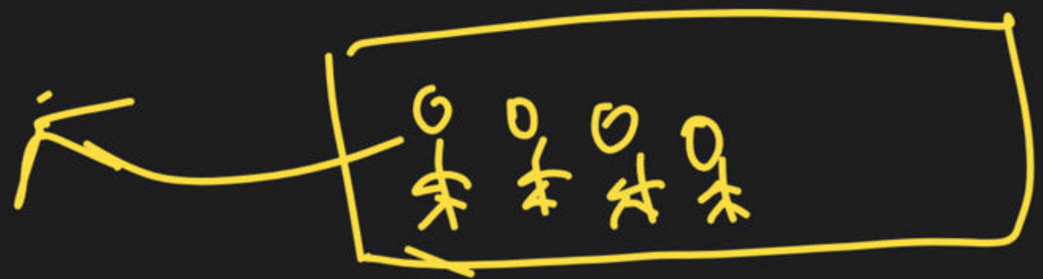
With Every element a priority is associated & elements are processed as per priority.

High priority will be processed first.

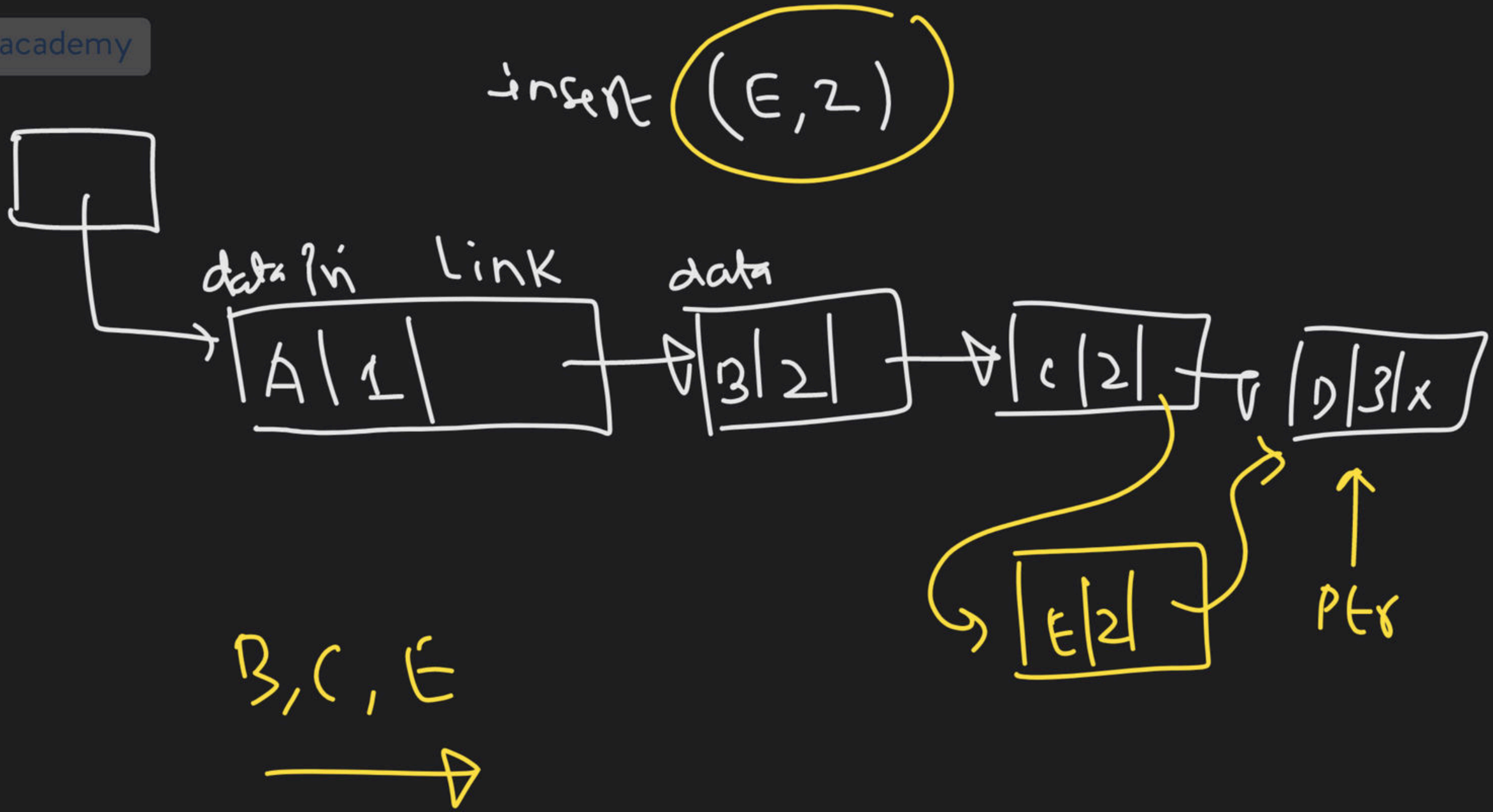
Same priority ⇒ insertion order.


Doc.
300/-
1 2 3 4 5 1
Queue
1060/-
Emergency

Doc.



All with same priority



High Priority → Small no.

Low Priority → Large no.

High priority → Large no.

Low priority → small no.

unacademy

Rohit

Priority Queue

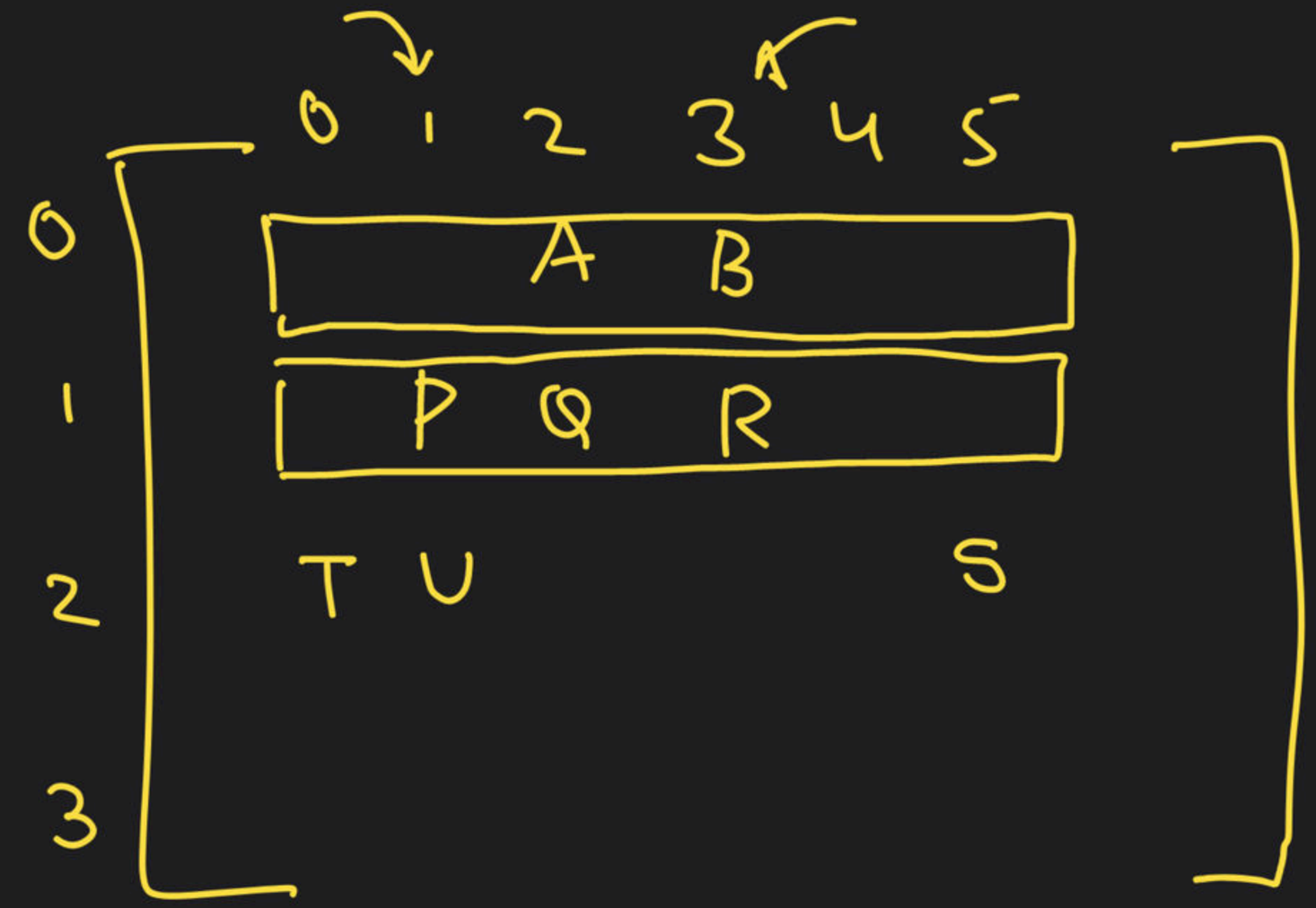
A[4][6];

unacademy

Front[2] = 5
Rear[2] = 1

Rear[4] ;

Front[4] ;



Front[0] = 2
↳ Rear[0] = 3

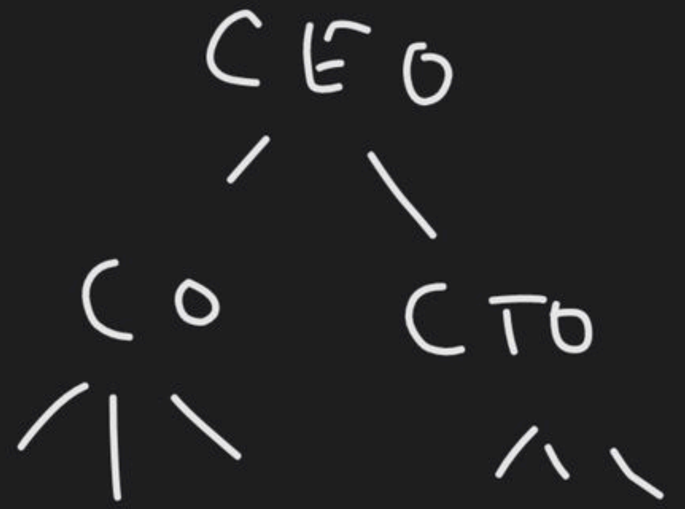
Front[i] : ⇒

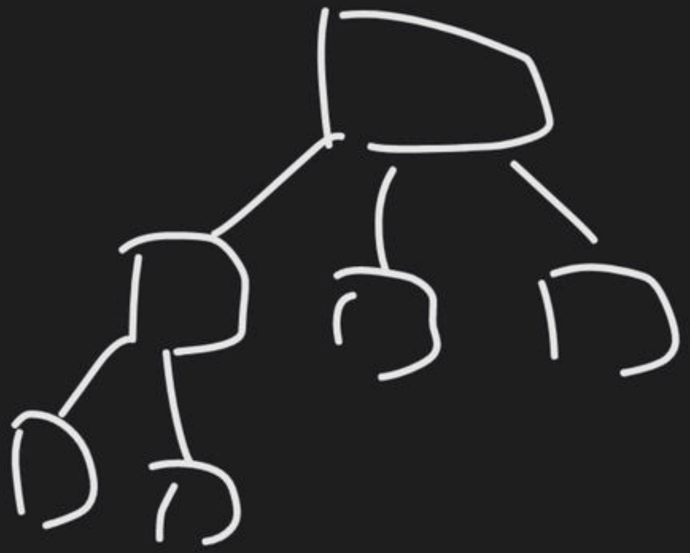
Front[1] = 1
Rear[1] = 3

Trees

Non-Linear data structure

1. Organization structure
2. Folder structure





3. HTML / XML (JSON Objects)
4. Binary Search Trees
5. Binary Heap
6. B-Trees, Bt-Trees
7. Parse Trees

unacademy

Trees

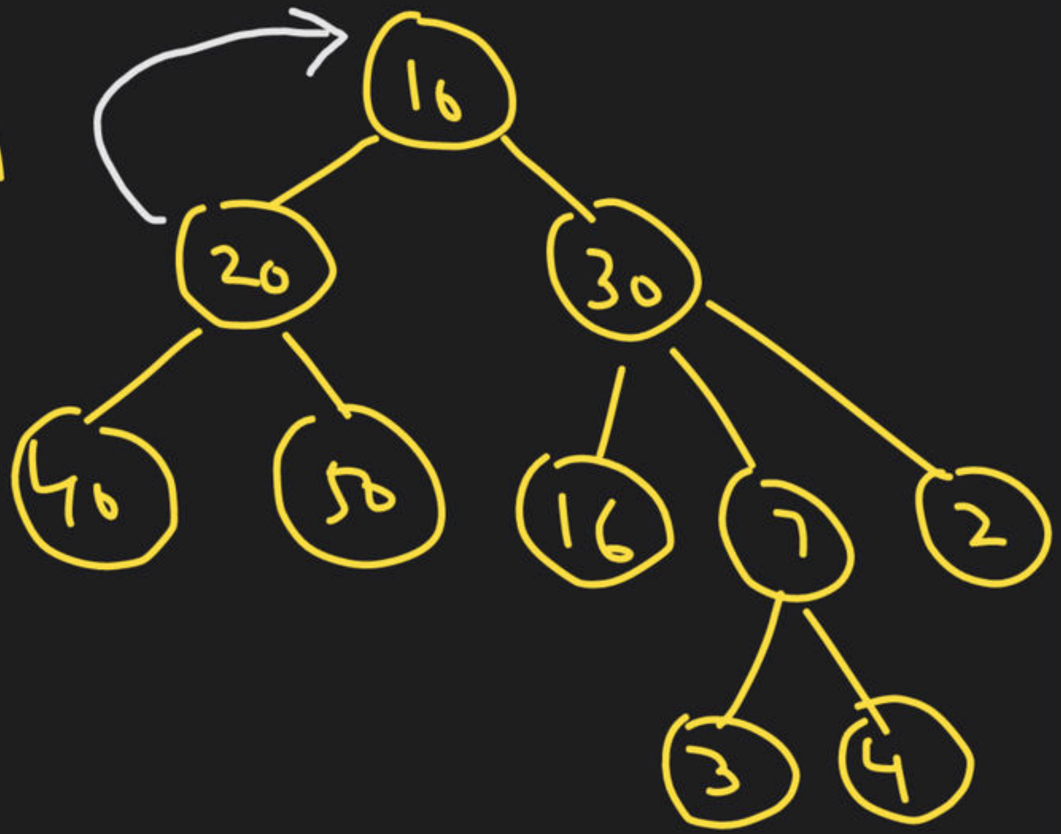
1. **Node**: Each element is represented by node

2. **child**: 20, 30 are childs of 10

   40, 50 are childs of 20

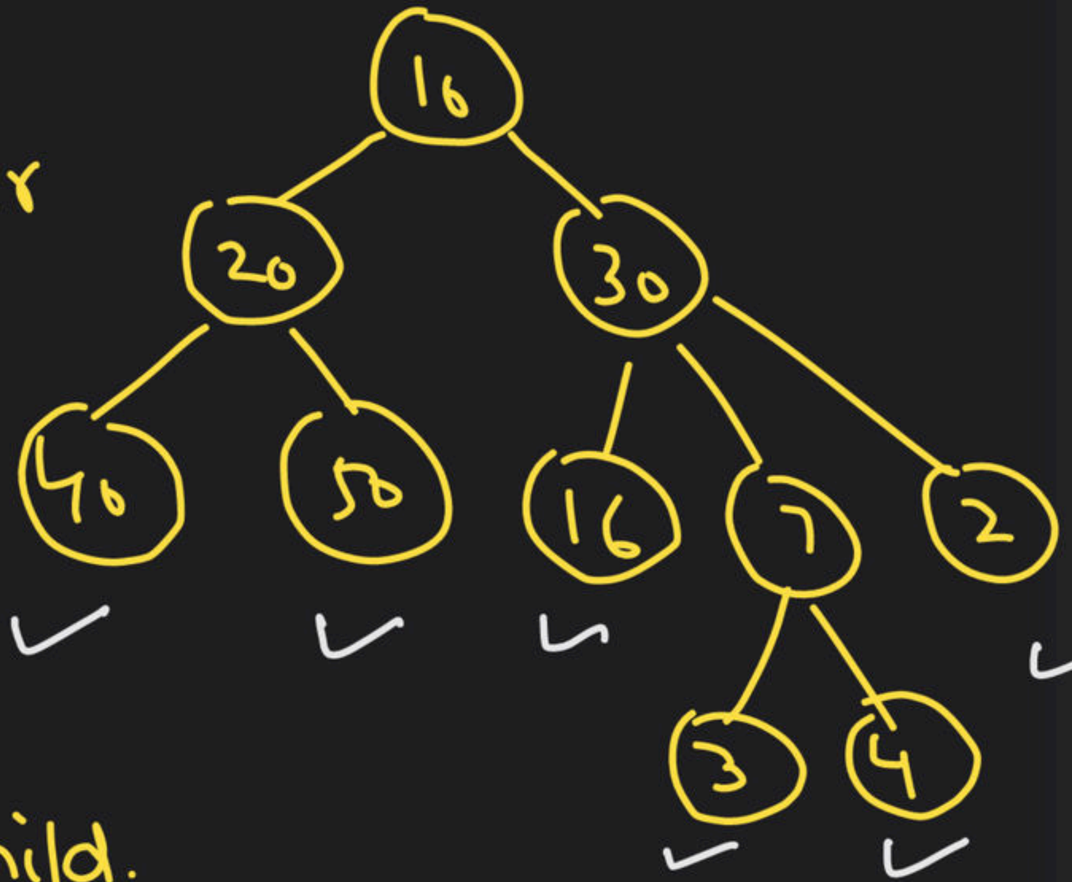
3. **Parent**: 20 is the parent of 40, 50

   10 is the parent of 20, 30



4.) **ROOT** : Distinguishable from other node (No parent)

10 is the root node



5.) **Leaf node** : Node with no child.

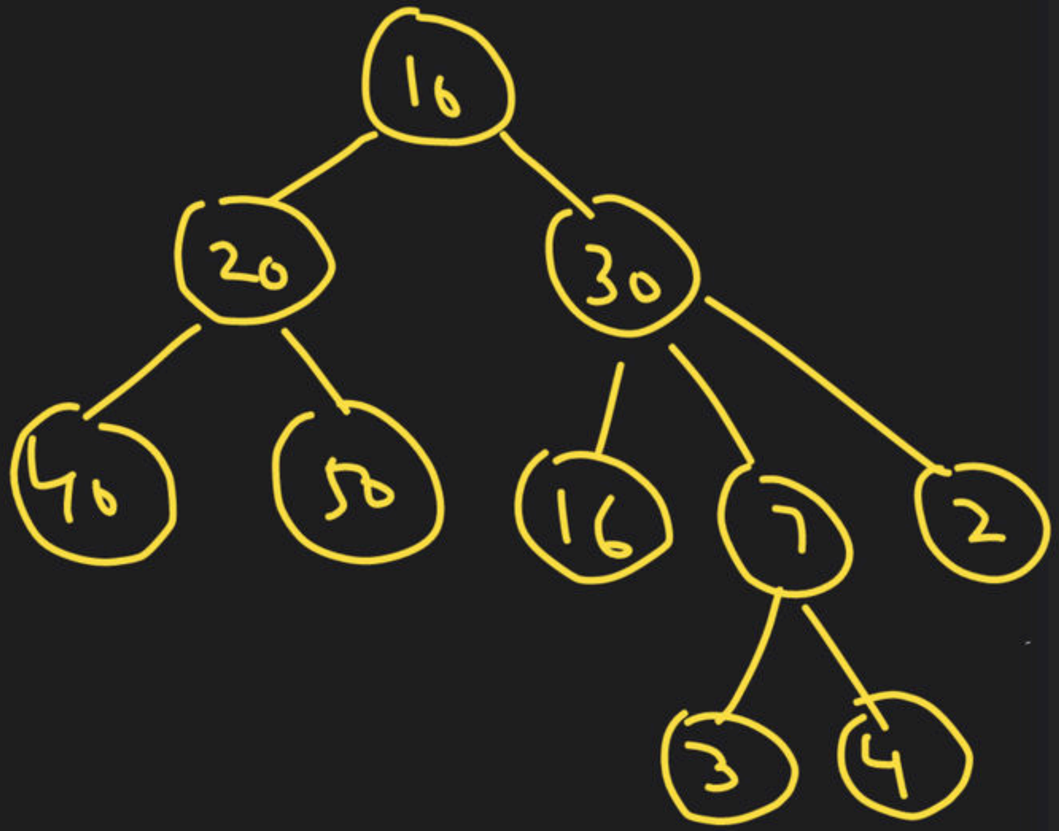
40, 50, 16, 3, 4, 2
all are leaf nodes.

6) Internal Node : Nodes with atleast 1 child

i,e 10, 20, 30, 7 are Internal nodes.

Trees/Graph

7) Degree of a Node: No. of childs



degree of a leaf node = 0

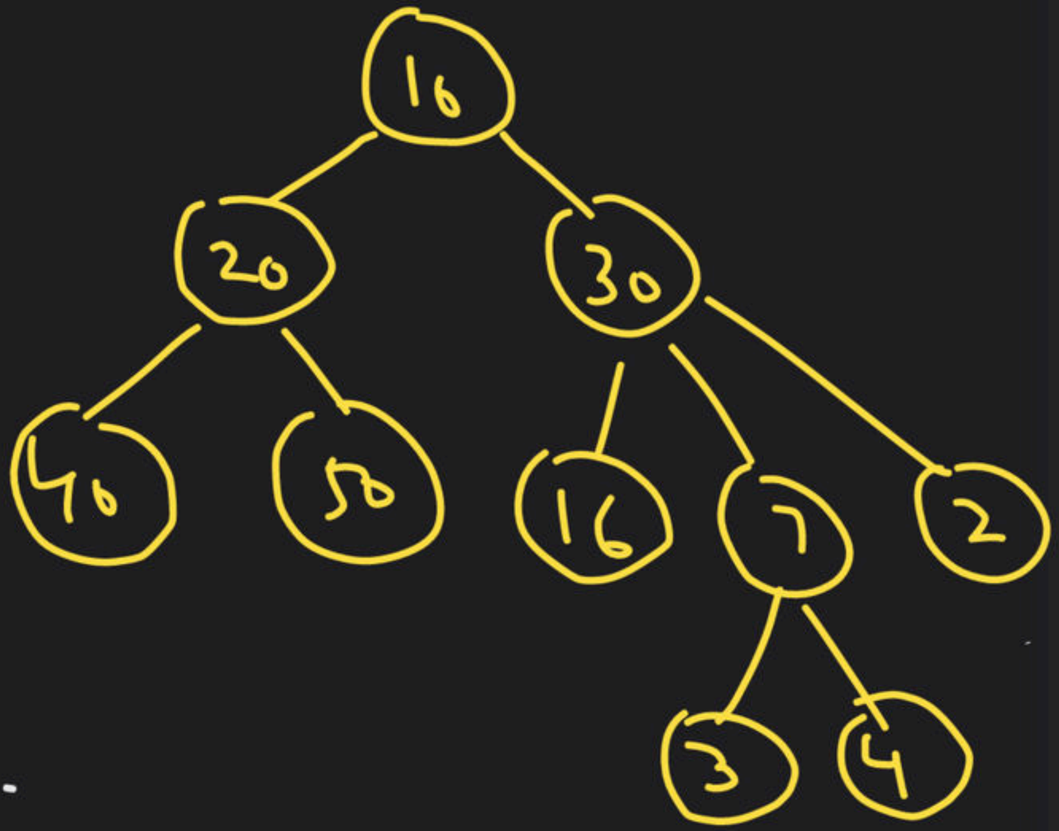
degree of a node with 1 child = 1

degree of a node with 2 childs = 2

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

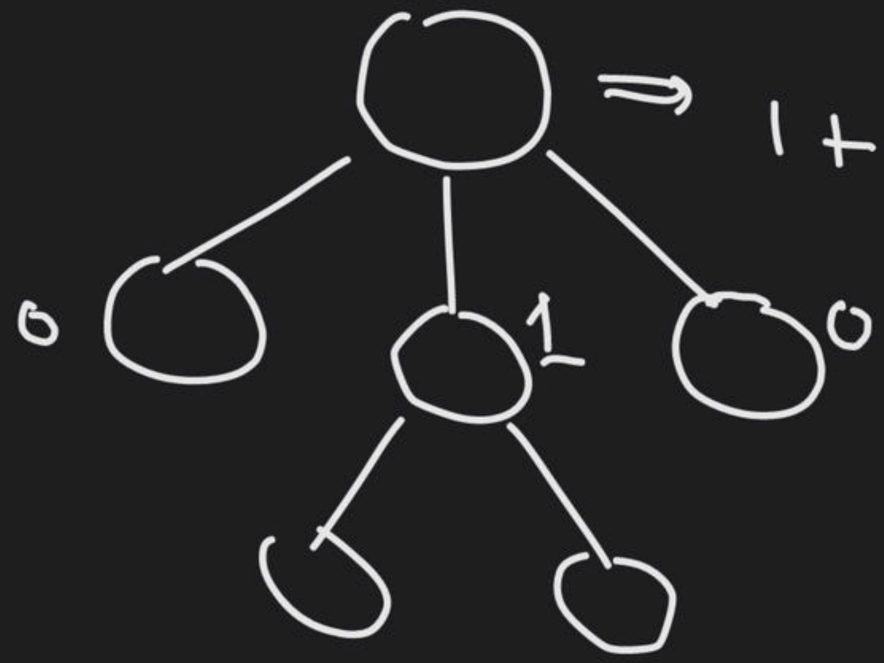
OR

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



unacademy

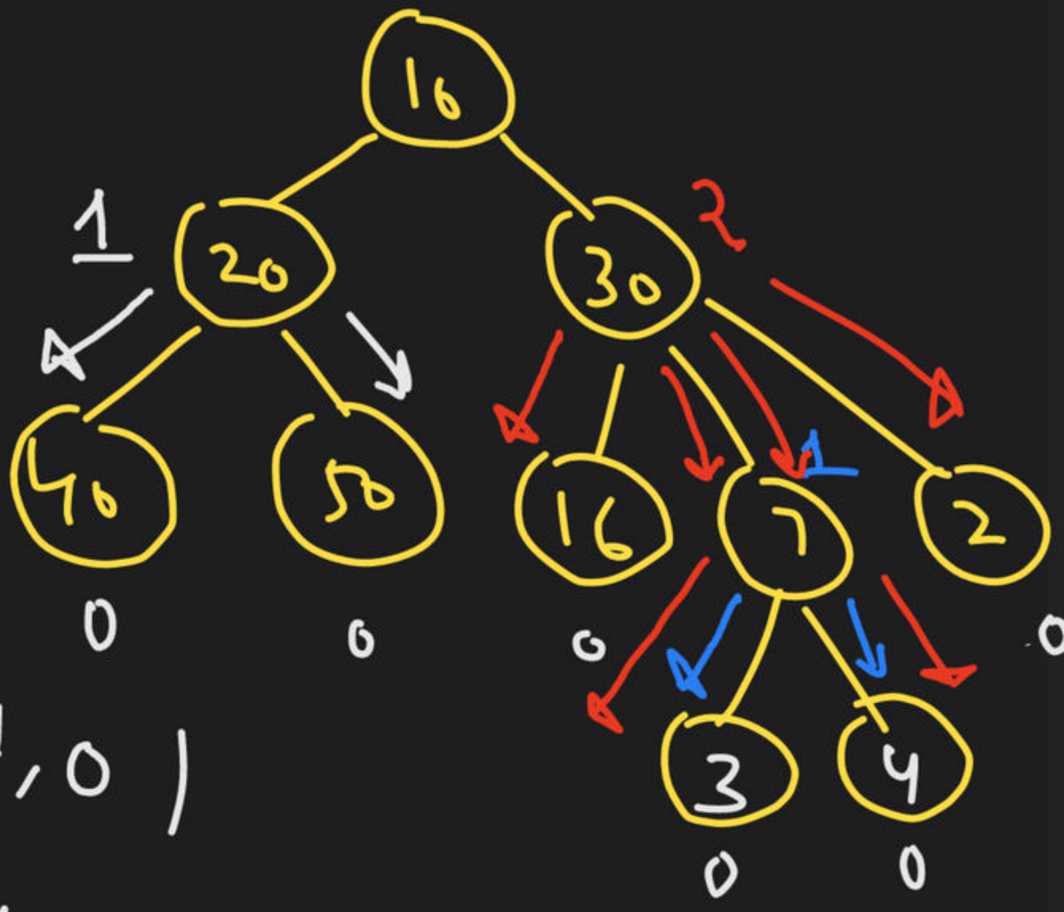
height of a leaf node = 0



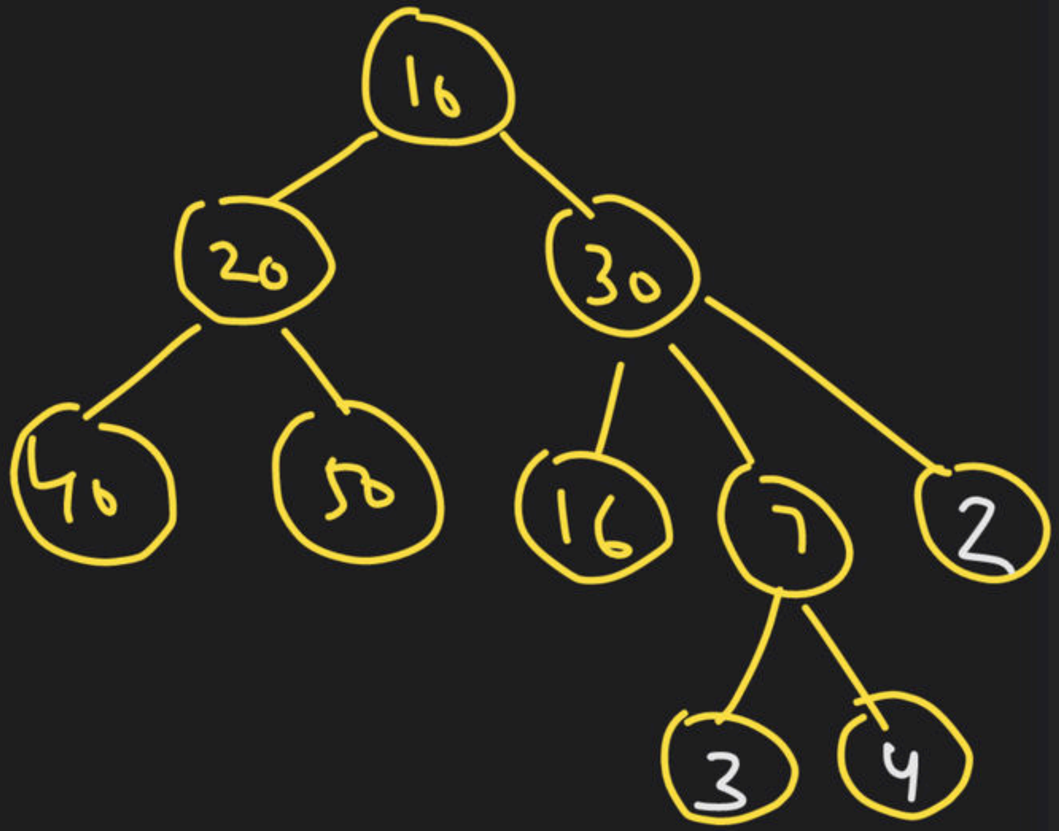
$\Rightarrow 1 + \max(0, 1, 0)$

$= 1 + 1$

$= 2$

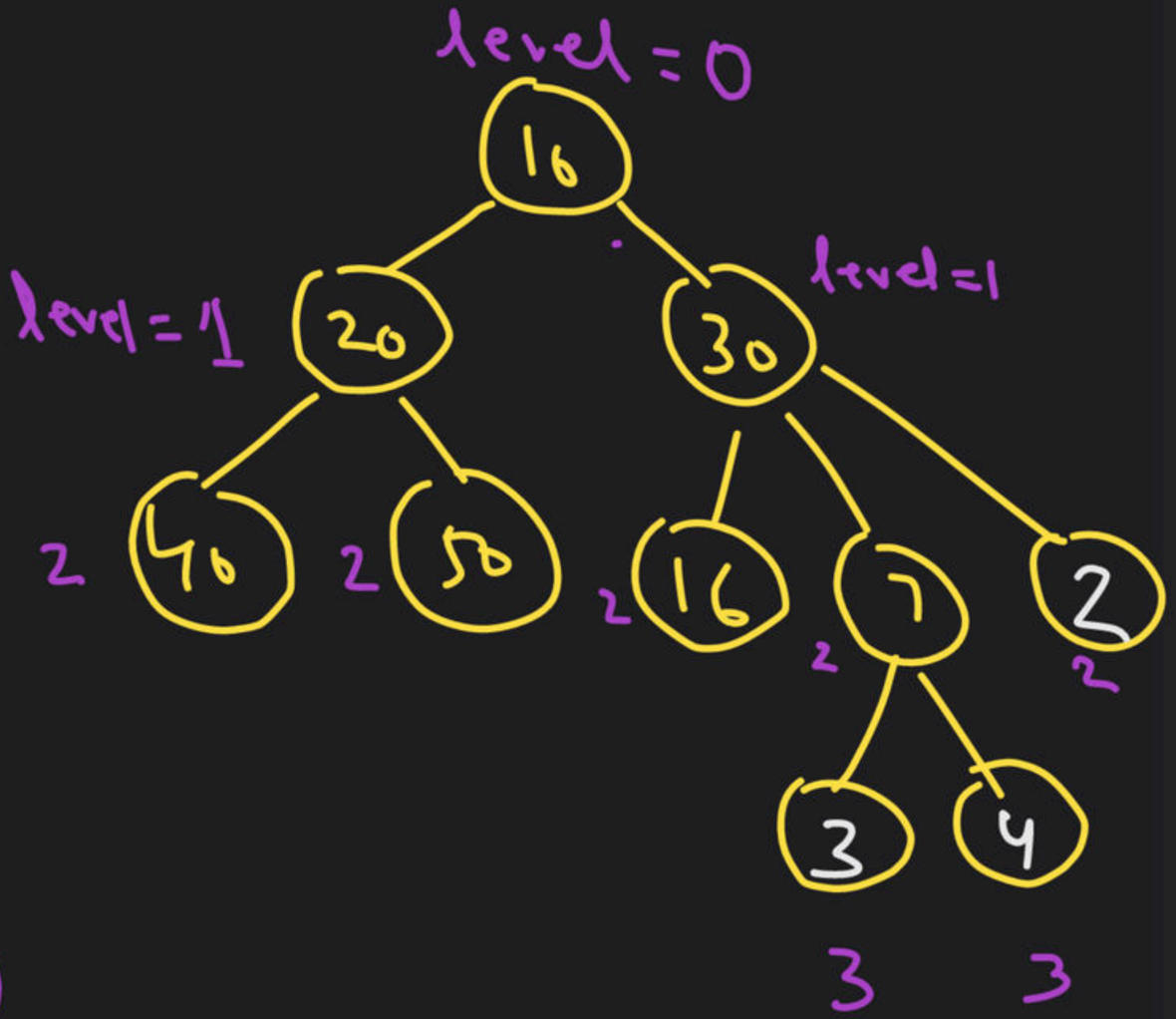


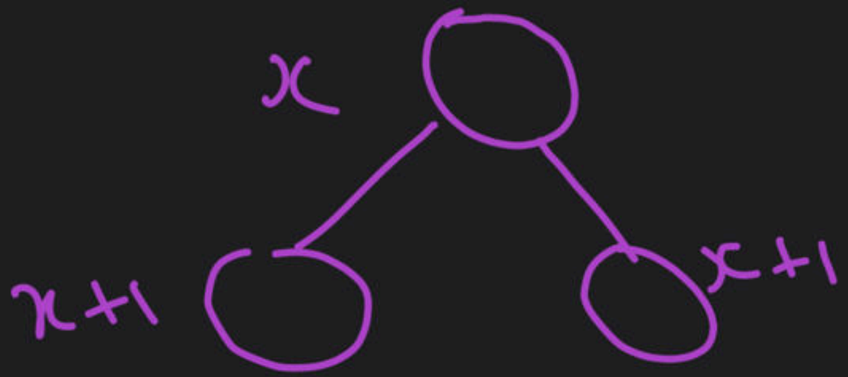
Height of tree = height of root node

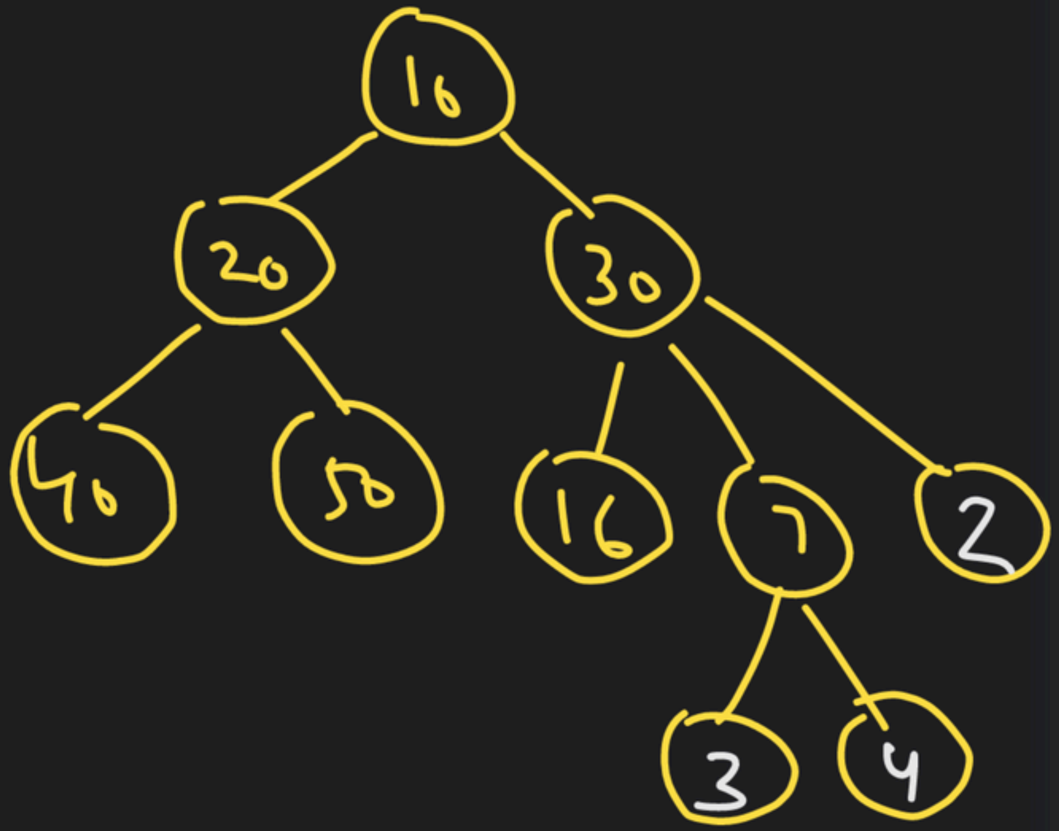


Depth/level of a node :

Level of a node X is the length of path from root node to node X.



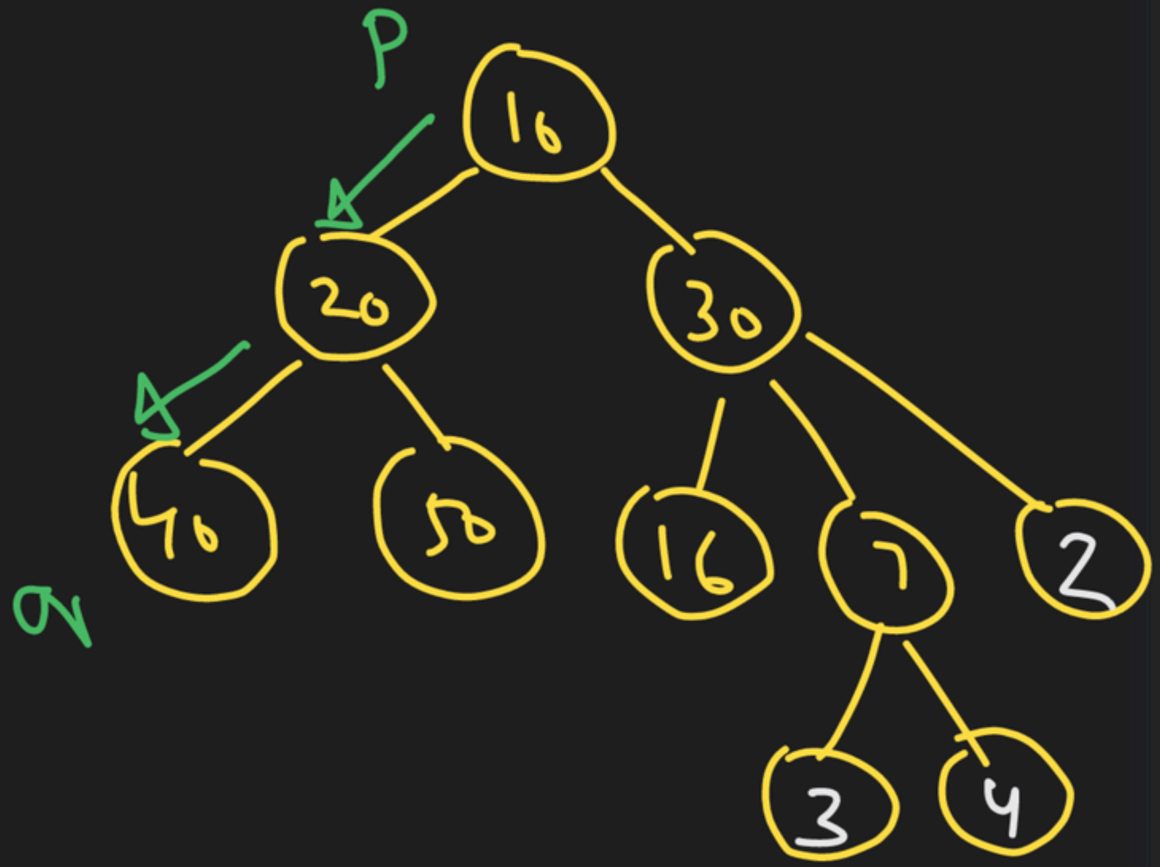




unacademy

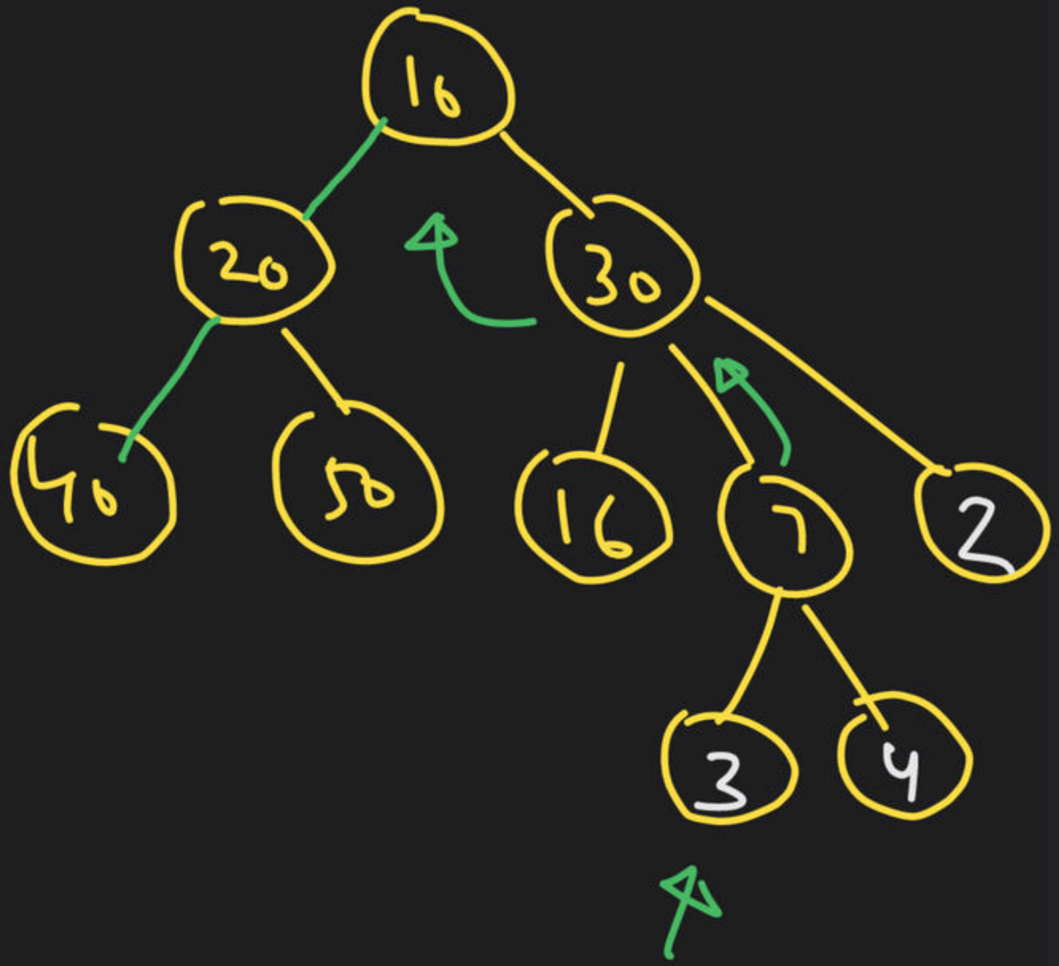
Ancestor of a node q :

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.



unacademy

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



7, 36, 10

unacademy

30 is an ancestor of

16, 7, 2, 3, 4

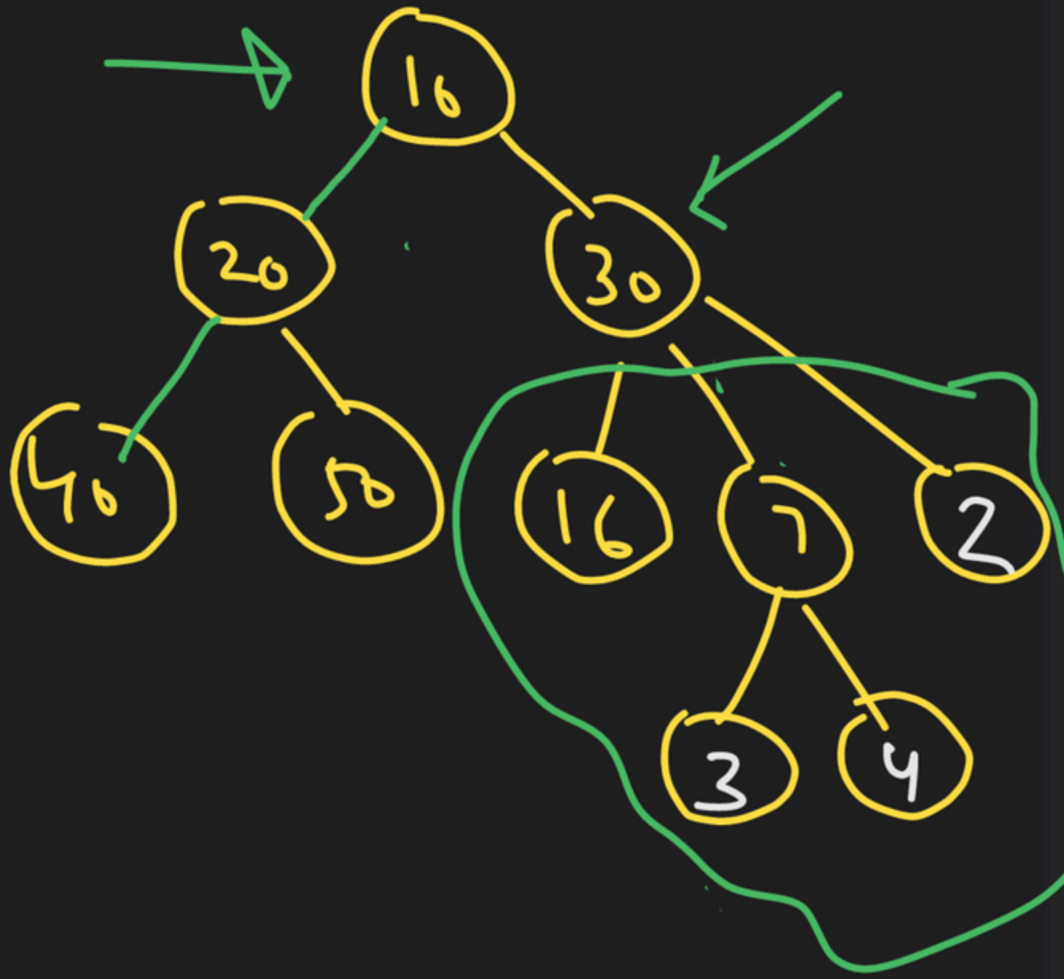
16 → is a descendent of 30

7 → " "

2 " "

3 " "

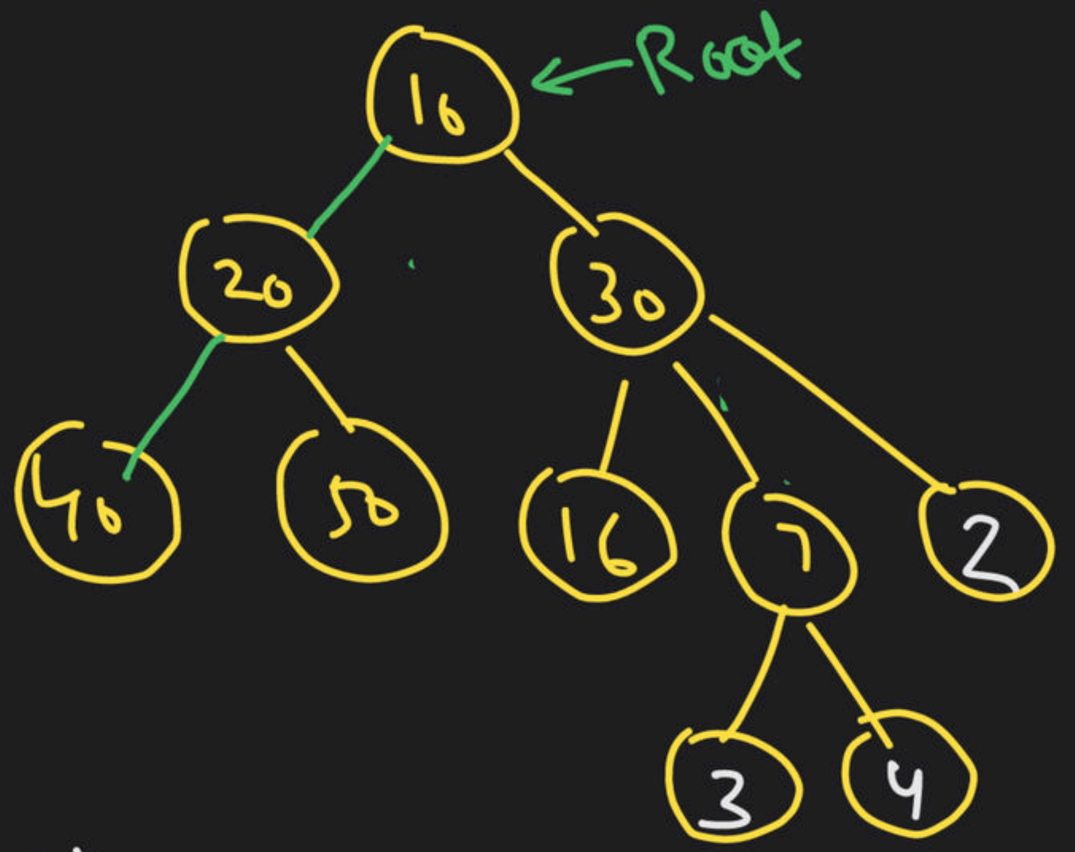
4



unacademy

Trees

Sibling: Nodes with same parent.



50, 16 ⇒ ~~Siblings~~

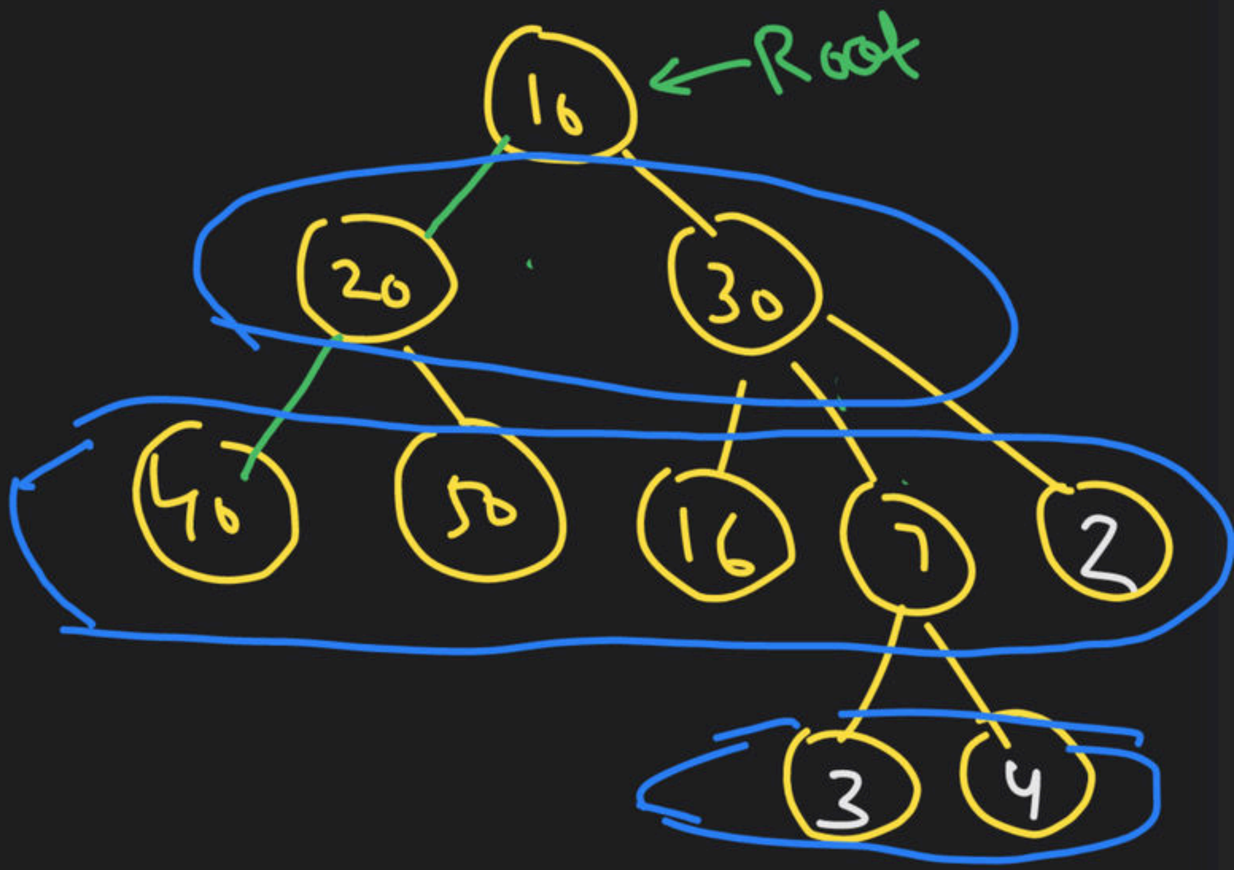
40, 50 ⇒ Siblings

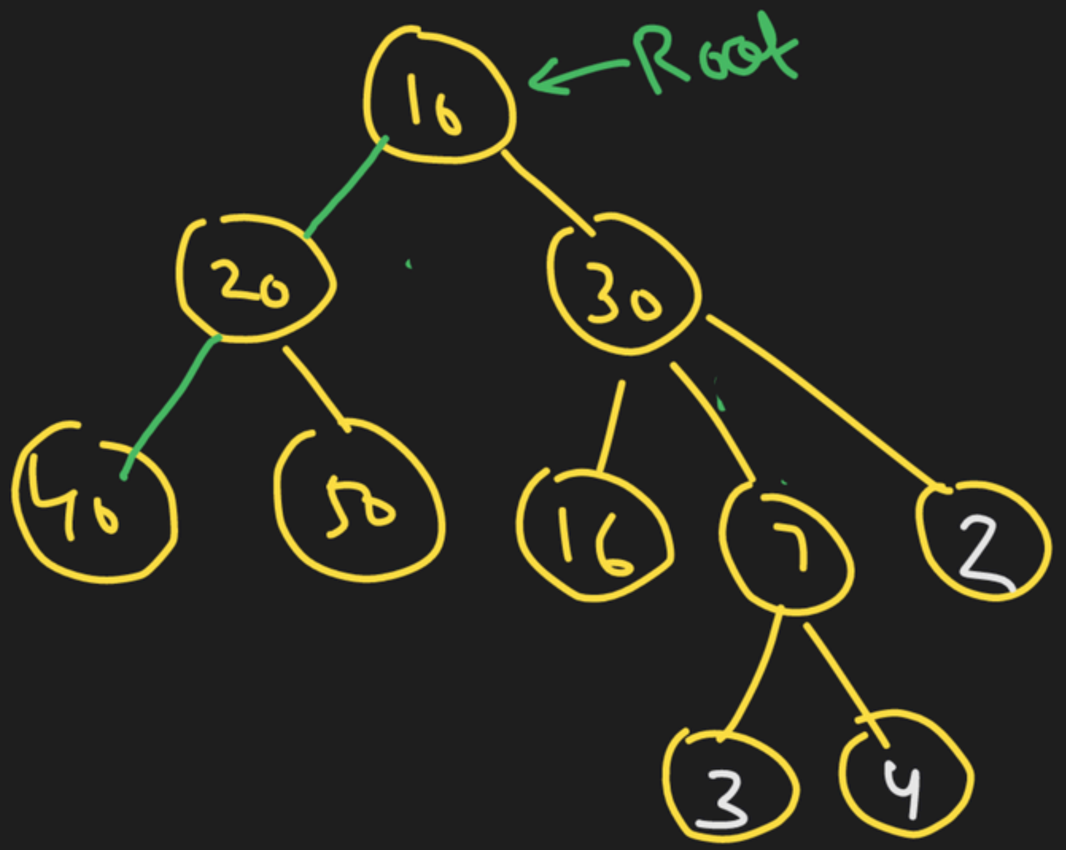
16, 7, 2 ⇒ Siblings

3, 4 ⇒ Siblings

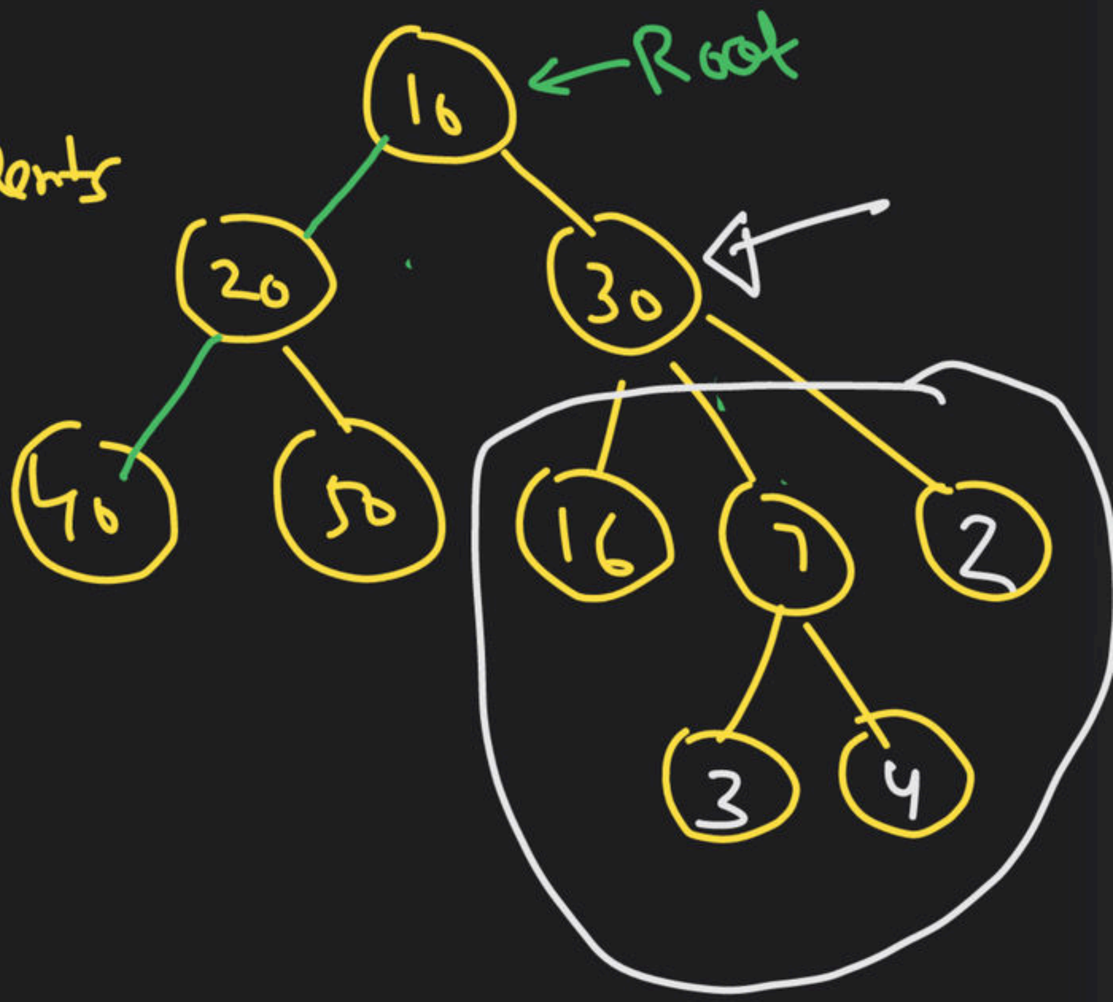
Generation:

All nodes at a particular level forms a Generation





Size of a node : No. of descendents of the node (including the node itself) is called as size of that node.



Size of node with key 30

$= 6\ (5+1)$

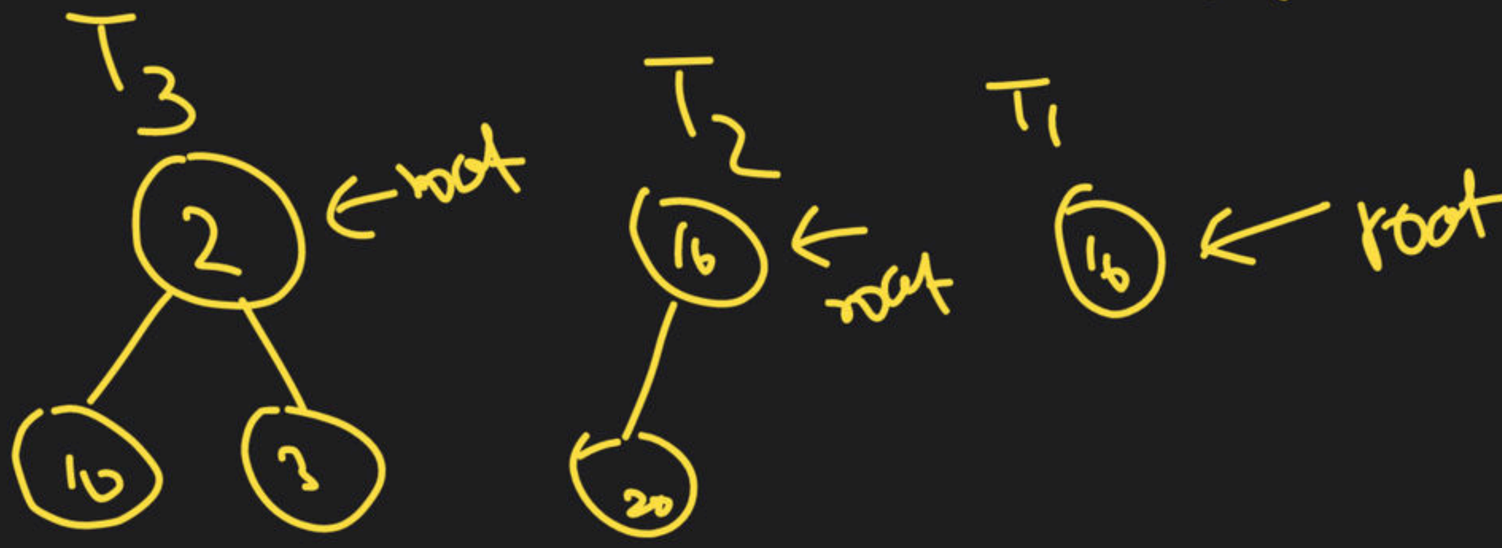
unacademy

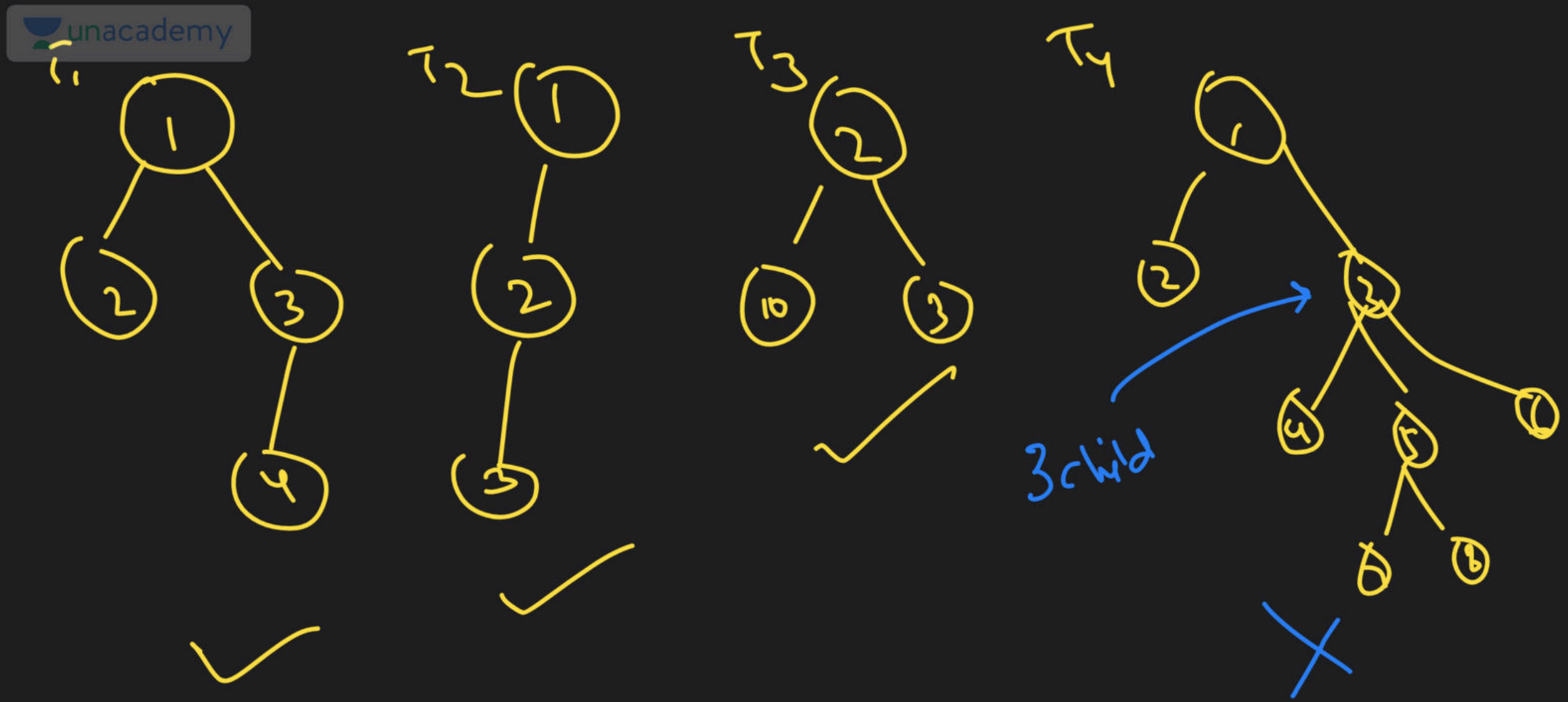
Binary Tree

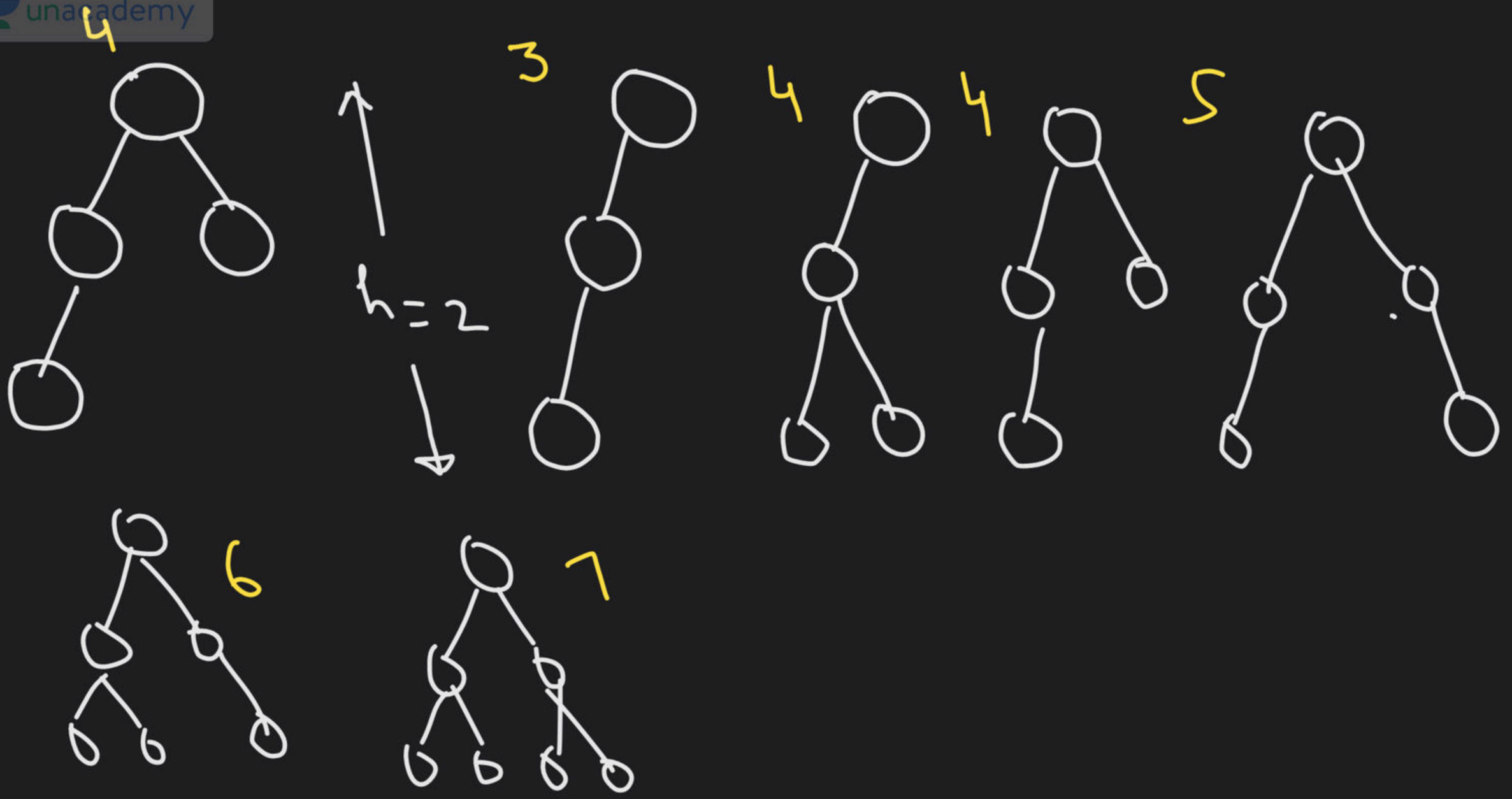
A node can have atmost 2 childs

0 child] leaf

1 child]
2-child] internal nodes

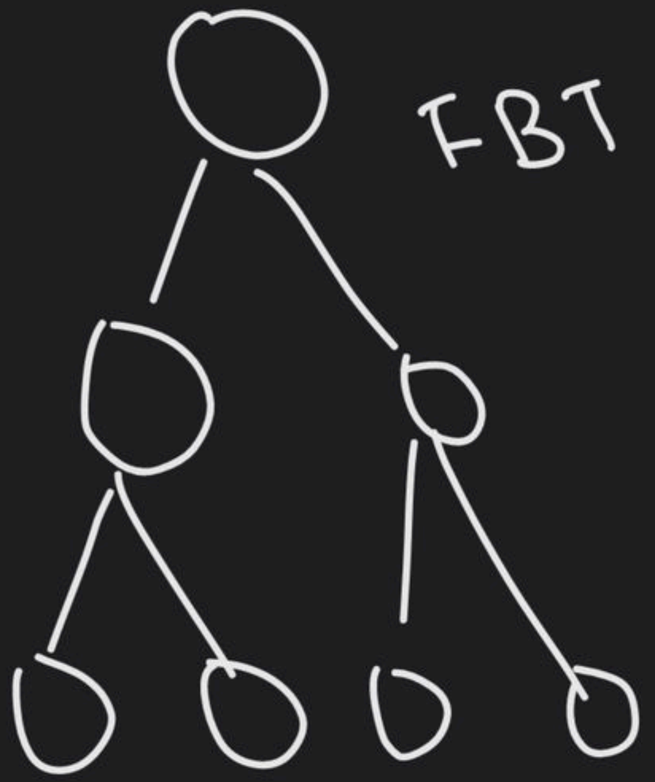



T1
1
2
3
4
T2
1
2
3
T3
2
10
3
T4
1
2
3
3 child
X



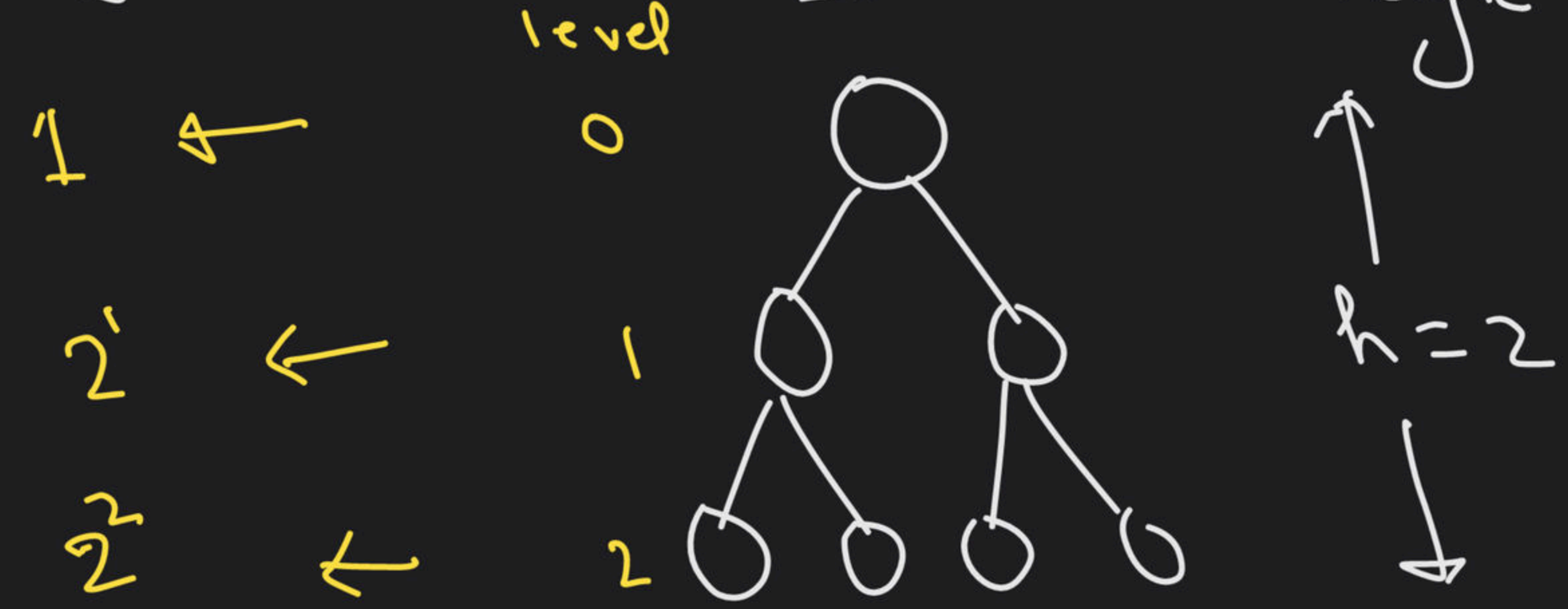
T

h = 2



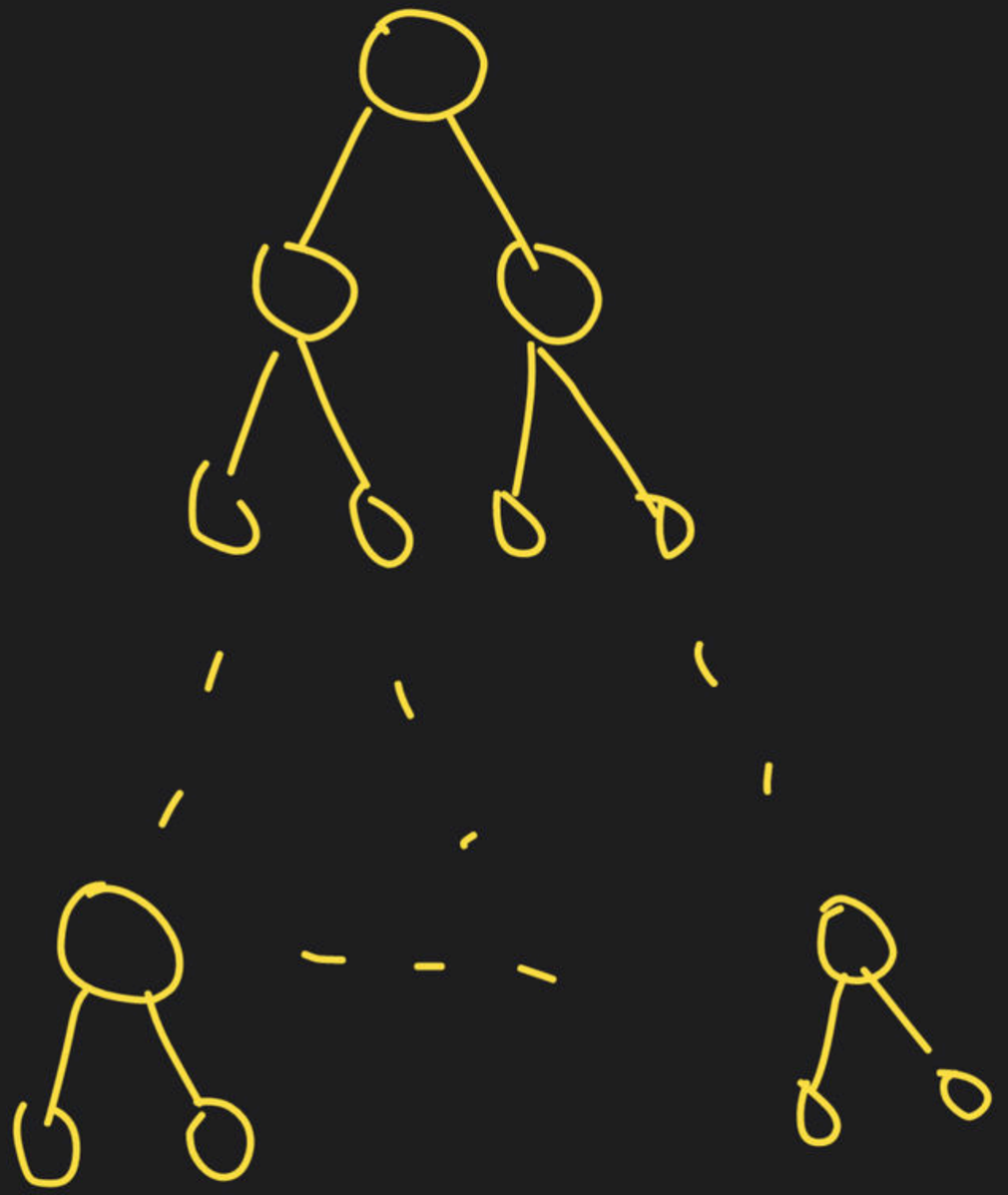
unacademy

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



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



$$N = 1 + 2^1 + 2^2 + \cdots + 2^h$$

(GP with $h+1$ terms)

$$= \frac{1\left(2^{h+1} - 1\right)}{2 - 1} = 2^{h+1} - 1$$

$$N = 2^{h+1} - 1$$

unacademy

$\cos(x+1)$

Expression Trees

unacademy

THANK YOU!

Here's to a cracking journey ahead!