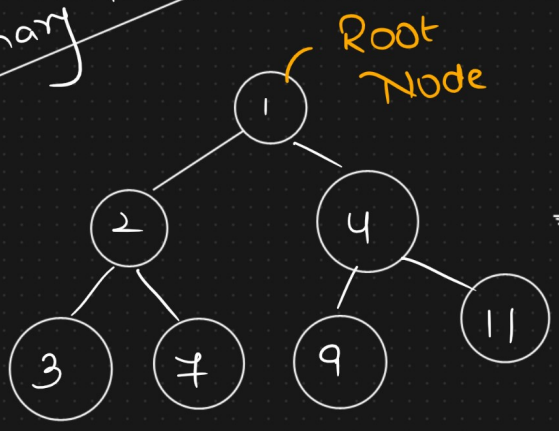
Revision of Traversal

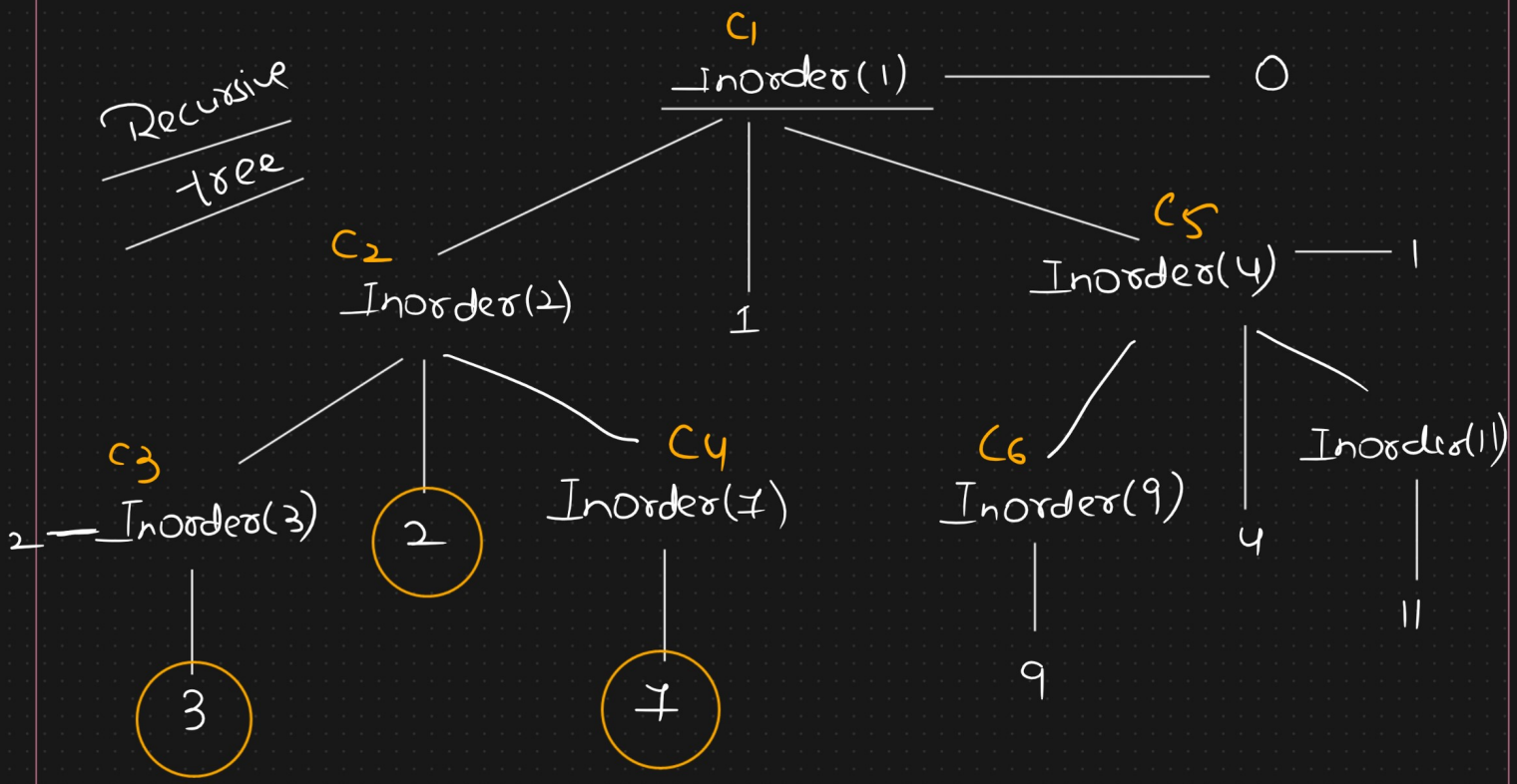
Inorder
- Left subtree
- Print (data)
- Right subtree

Binary Tree



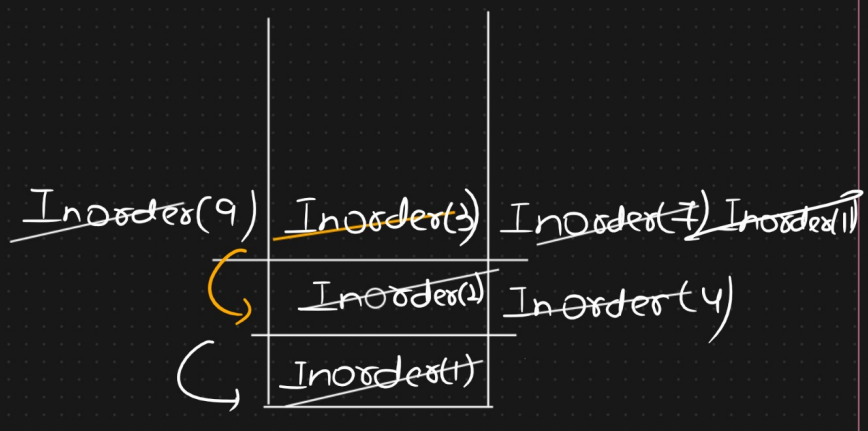
⇒ 3, 2, 7, 1, 9, 4, 11

Recursive tree



3, 2, 7, 1, 9, 4, 11

Levels = 3



Stack Space = # Levels

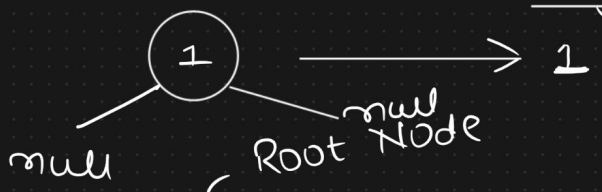
= Balanced tree

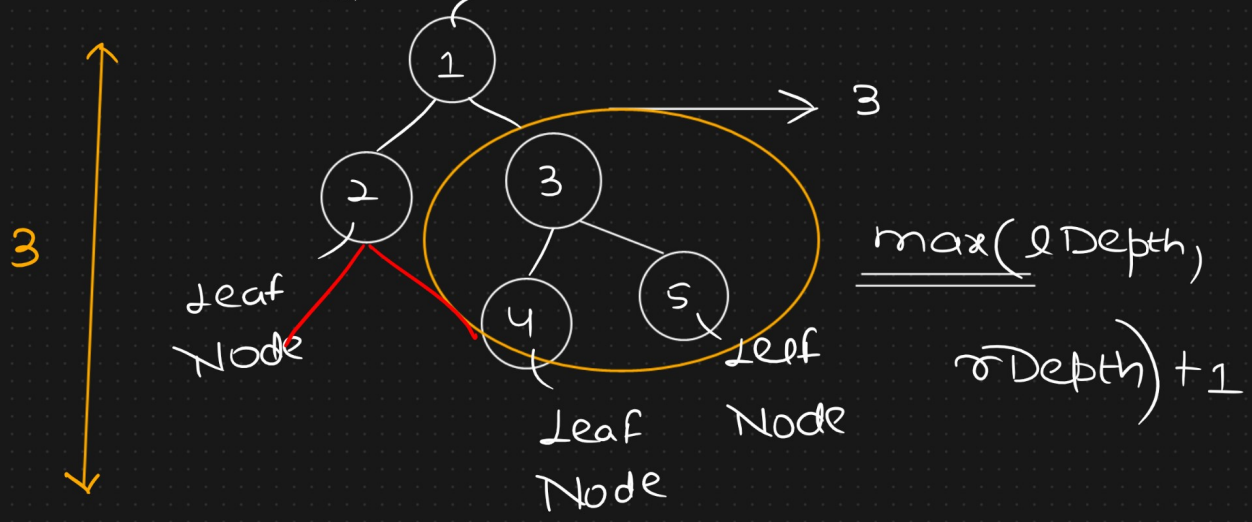
$= \log_2 n$

1

Height of Tree

Height of tree





max(lDepth, rDepth) + 1

maxDepth(Node node)

1 + max(lDepth, rDepth)

maxDepth(node.right)

= 3

maxDepth(node.left)

maxDepth(2)

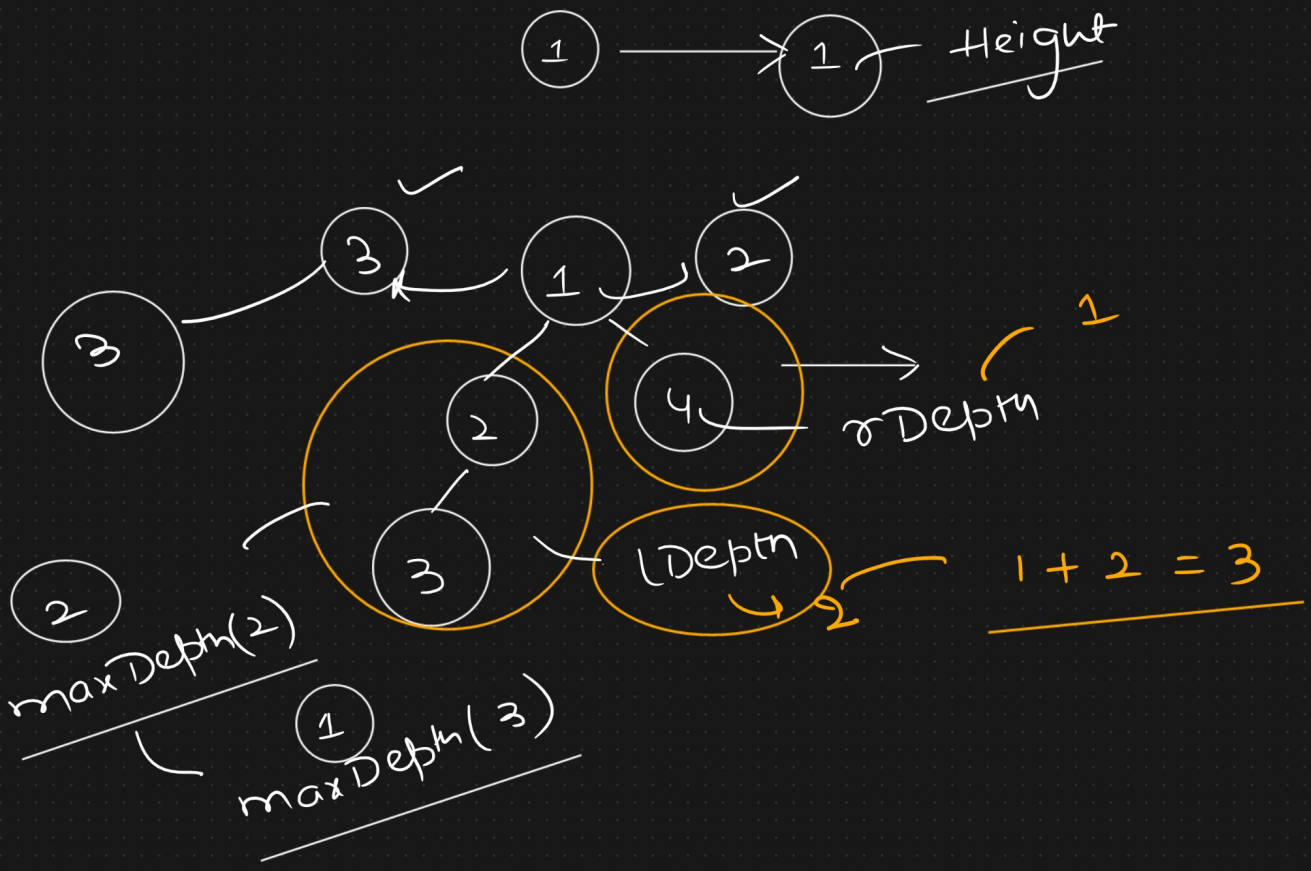
1 +

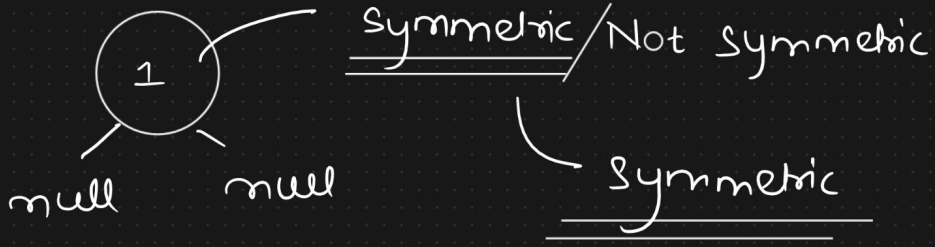
max(lDepth, rDepth)

Root — Null

↳ Empty tree

↳ 0



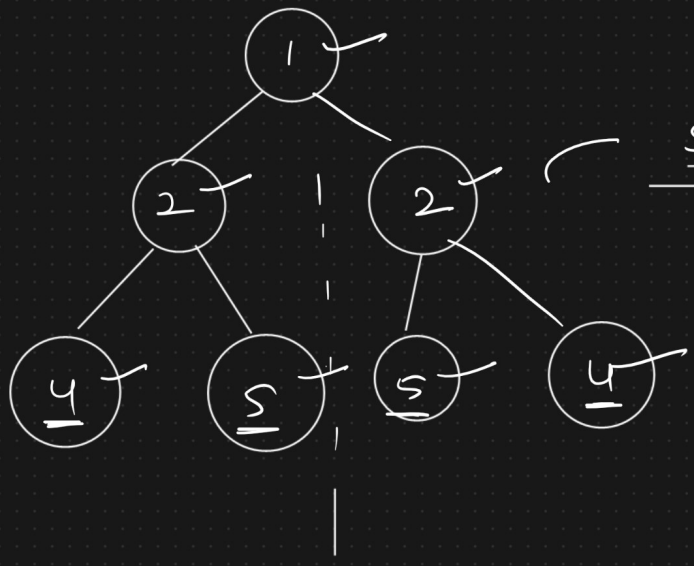


t1 = 1 , t2 = 1

true

isMirror(Null, Null) && true

isMirror(Null, Null)



Symmetric tree

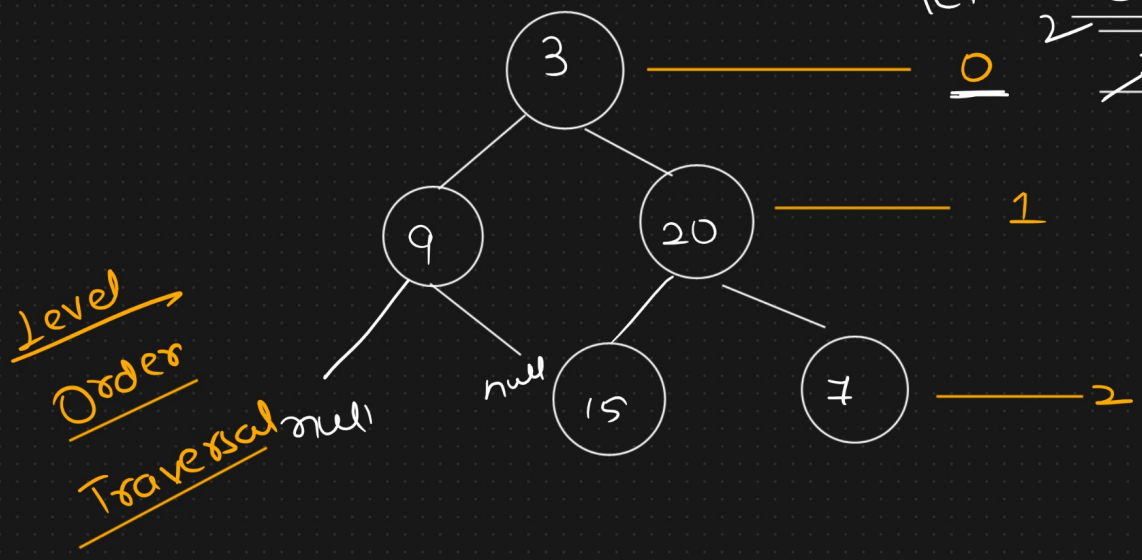
↓

only one time traversal of all the nodes

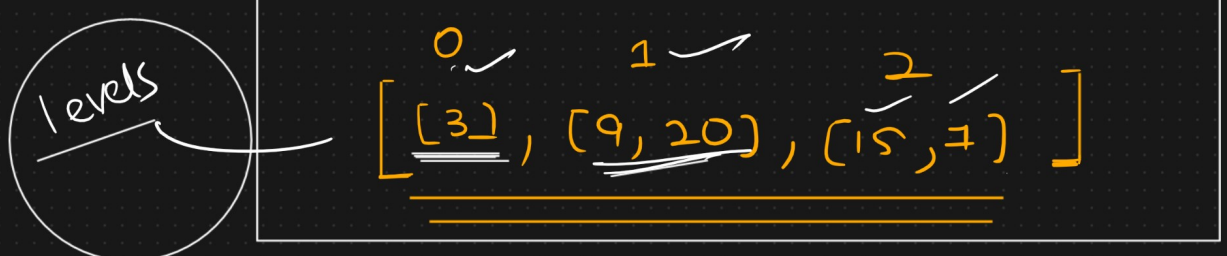
⇓

$O(n)$

Space complexity → $O(n)$



Level Order Traversal

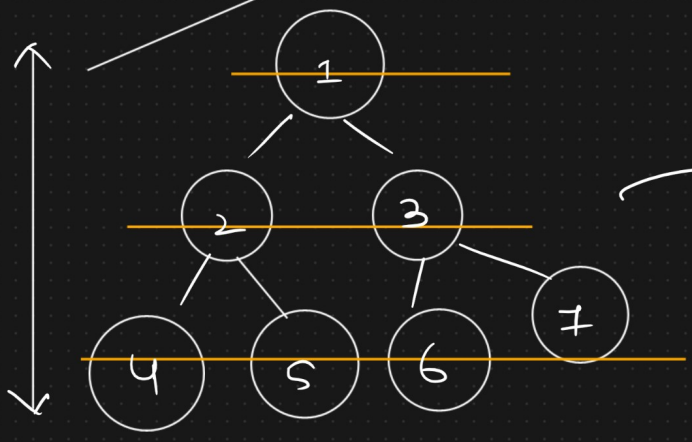


Graph

- Dfs
  - Preorder
  - Postorder
  - Inorder
- Bfs
  - Level order traversal

Depth first Traversal

- Recursion
- Stack
- Data Structure



1, 2, 4, 5, 3, 6, 7

Preorder

Breadth first Traversal → 1, 2, 3, 4, 5, 6, 7

Level Order Traversal

Queue

Every tree is a graph

↓

Every graph is not a tree

Tree → acyclic — Not contain any cycle

Graph → can be cyclic & acyclic