

# Maximum Depth of Binary Tree

## Key Idea:

The maximum depth is simply:

$$\text{max depth} = 1 + \max(\text{depth of left subtree, depth of right subtree})$$

- If the node is null, the depth is 0.
- Otherwise, recursively calculate depths for left and right subtrees.

## Approaches:

### 1. Recursive DFS (Top-down):

- Base case: if root is null, return 0.
- Recursively compute depths of left and right sub-trees.
- Return  $1 + \max(\text{leftDepth}, \text{rightDepth})$ .

### 2. Iterative BFS (Level order):

- Traverse level by level using a queue.
- Count levels until queue is empty.
- Depth = number of levels.

## Complexity:

- Time:  $O(N)$  - visit each node once.
- Space:  $O(H)$  for recursion ( $H = \text{height}$ ) or  $O(N)$  for BFS queue (worst case).