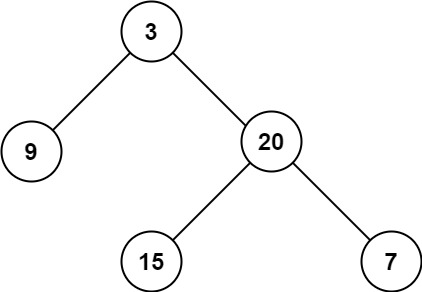
# **Maximum Depth of Binary Tree**

Given the root of a binary tree, return *its maximum depth*.

A binary tree's **maximum depth** is the number of nodes along the longest path from the root node down to the farthest leaf node.

**Example 1:**



**Input:** root = [3,9,20,null,null,15,7]

**Output:** 3

**Example 2:**

**Input:** root = [1,null,2]

**Output:** 2

**Constraints:**

* The number of nodes in the tree is in the range [0, 104].
* -100 <= Node.val <= 100

/\*\*

\* Definition for a binary tree node.

\* public class TreeNode {

\* public int val;

\* public TreeNode left;

\* public TreeNode right;

\* public TreeNode(int val=0, TreeNode left=null, TreeNode right=null) {

\* this.val = val;

\* this.left = left;

\* this.right = right;

\* }

\* }

\*/

public class Solution {

int maxVal = 0;

public int MaxDepth(TreeNode root) {

Traverse(root, 1);

return maxVal;

}

void Traverse(TreeNode root, int depth)

{

if(root == null)

{

return;

}

if(depth > maxVal)

{

maxVal = depth;

}

depth ++;

Traverse(root.left, depth);

Traverse(root.right, depth);

}

}