

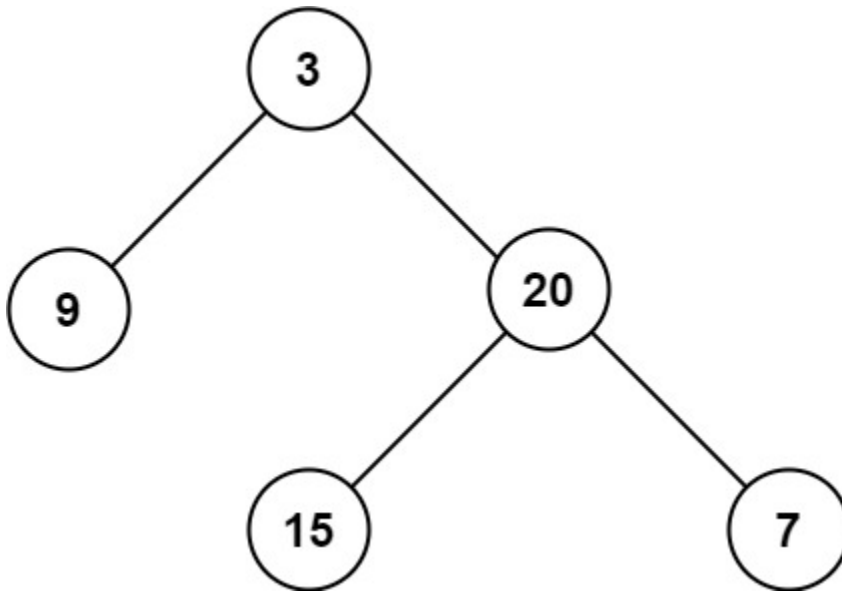
Leetcode Problem 1. (Easy)

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, 10].
- $-100 \leq \text{Node.val} \leq 100$

Link: <https://leetcode.com/problems/maximum-depth-of-binary-tree/>

```
class Solution {  
    public int maxDepth(TreeNode root) {
```

```

    if (root == null) {
        return 0;
    }
    int leftDepth = maxDepth(root.left);
    int rightDepth = maxDepth(root.right);
    return 1 + Math.max(leftDepth, rightDepth);
}
}

```

<

Problem List

>

Premium

🕒 0

👤

Description

Editorial

Solutions (9.2K)

Submissions

Accepted

Next question

105. Construct Binary Tree from Preorder and Inorder Traversal

More challenges

110. Balanced Binary Tree

111. Minimum Depth of Binary Tree

559. Maximum Depth of N-ary Tree

All statuses

All languages

Accepted

a few seconds ago

Java

Sakib Rahman

Apr 24, 2023 11:54

Details

+ Solution

Java

Runtime 0 ms

Beats 100%

Memory 43.1 MB

Beats 6.44%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

```

/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }

```

Console

Run

Submit