

Leetcode Problem 1. (Easy)

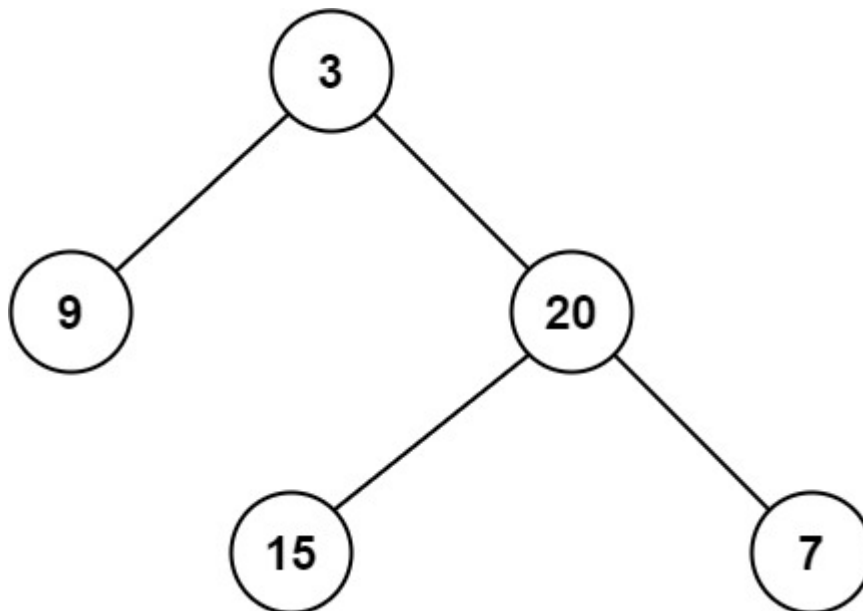
Minimum Depth of Binary Tree

Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

Note: A leaf is a node with no children.

Example 1:



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

Output: 2

Example 2:

Input: root = [2,null,3,null,4,null,5,null,6]

Output: 5

Constraints:

- The number of nodes in the tree is in the range [0, 10⁵].
- -1000 ≤ Node.val ≤ 1000

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

```

class Solution {
    public int minDepth(TreeNode root) {
        if (root == null) {
            return 0;
        }
        if (root.left == null && root.right == null) {
            return 1;
        }
        int minDepth = Integer.MAX_VALUE;
        if (root.left != null) {
            minDepth = Math.min(minDepth(root.left), minDepth);
        }
        if (root.right != null) {
            minDepth = Math.min(minDepth(root.right), minDepth);
        }
        return minDepth + 1;
    }
}

```

LeetCode

Problem List

Premium

0

Description Editorial Solutions (4.2K) Submissions

Accepted

Next question

112. Path Sum

More challenges

102. Binary Tree Level Order Traversal

All statuses All languages

Accepted a few seconds ago Java

Sakib Rahman Apr 26, 2023 23:10

Details + Solution

Java

Runtime 17 ms Beats 5.23% Memory 63 MB Beats 25.20%

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