Minimum Depth of Binary Tree

*Description:*

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:*

Leaf node is a node with no children.

*Instances:*

|  |  |  |
| --- | --- | --- |
|  | | *minimum depth = 0* |
|  | | *minimum depth = 1* |
|  |  | *minimum depth = 2* |
|  | | *minimum depth = 2* |

*Code – Recursive:*

*/\*\**

*\* Definition for a binary tree node.*

*\* struct TreeNode {*

*\* int val;*

*\* TreeNode \*left;*

*\* TreeNode \*right;*

*\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}*

*\* };*

*\*/*

*class Solution {*

*public:*

*int minDepth(TreeNode\* root) {*

*return minDepth(root, false);*

*}*

*int minDepth(TreeNode\* root, bool hasbrother) {*

*if (!root)*

*return hasbrother ? INT\_MAX : 0;*

*return min(minDepth(root->left, root->right != NULL),*

*minDepth(root->right, root->left != NULL)) + 1;*

*}*

*};*