104. Maximum Depth of Binary Tree

Easy

112746FavoriteShare

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

Note: A leaf is a node with no children.

Example:

Given binary tree [3,9,20,null,null,15,7],

3  
 / \  
 9 20  
 / \  
 15 7

return its depth = 3.

Accepted

453,389

Submissions

766,990

/\*\*

\* 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 maxDepth(TreeNode\* root) {

if(root==NULL) return 0;

queue<TreeNode\*> q;

q.push(root);

int height=0;

int childCount=1;

while(!q.empty()&&childCount!=0){

int c=0;

for(int i=0;i<childCount;i++){

TreeNode\* curr = q.front();

q.pop();

if(curr->left!=NULL){

q.push(curr->left);

c++;

}

if(curr->right!=NULL){

q.push(curr->right);

c++;

}

}

childCount=c;

height++;

}

return height;

}

};

Success

[Details](https://leetcode.com/submissions/detail/210859248/)

Runtime: 16 ms, faster than 100.00% of C++ online submissions forMaximum Depth of Binary Tree.

Memory Usage: 19.3 MB, less than 79.97% of C++ online submissions forMaximum Depth of Binary Tree.