



链滴

# [每日 LeetCode] 104. Maximum Depth of Binary Tree

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1558361943696>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 104. Maximum Depth of Binary Tree](#)

### Description:

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]`,



return its depth = 3.

---

思路：本题要求二叉树的最大深度。简单的可使用递归法，如果二叉树为空，则深度为0 如果不为空分别求左子树的深度和右子树的深度，取最大的再加1。也可使用非递归，采用BFS非递归方法实现。体如下。

---

### C++代码（递归）

```
/**
 * 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)
            return 0;
        else {
            int left = maxDepth(root->left);
            int right = maxDepth(root->right);
            return left>right?left+1:right+1;
        }
    }
};
```

运行时间: 12ms

运行内存: 19.3M

---

C++代码 (非递归)

```
/**
 * 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;
        int res = 0;
        queue<TreeNode*> q;
        q.push(root);
        while(!q.empty())
        {
            ++ res;
            for(int i = 0, n = q.size(); i < n; ++ i)
            {
                TreeNode *p = q.front();
                q.pop();

                if(p -> left != NULL)
                    q.push(p -> left);
                if(p -> right != NULL)
                    q.push(p -> right);
            }
        }
        return res;
    }
};
```

运行时间: 4ms

运行内存: 19.3M