

## (1.6) LC104.二叉树的最大深度

[104. 二叉树的最大深度 - 力扣 \(LeetCode\)](#)

### 法一：层序遍历

思路：二叉树的深度和层序遍历的思想比较相似，因为二叉树有几层实际上就是深度为几。

代码：

```
class Solution {
public:
    int maxDepth(TreeNode* root) {
        //利用层序遍历做
        int depth = 0;
        queue<TreeNode*> que;
        if(root) que.push(root);

        while(!que.empty()){
            depth++;
            int size = que.size();
            for(int i = 0; i < size; i++){
                TreeNode* tmp = que.front();
                que.pop();
                if(tmp->left) que.push(tmp->left);
                if(tmp->right) que.push(tmp->right);
            }
        }
        return depth;
    }
};
```

### 法二：递归

代码：

```
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if(root == NULL) return 0;
        return max(maxDepth(root->left), maxDepth(root->right)) + 1;
    }
}
```

