

二叉树的深度

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518919190306>

来源网站: [链滴](#)

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

题目描述

输入一棵二叉树，求该树的深度。从根结点到叶结点依次经过的结点（含根、叶结点）形成树的一条路径，最长路径的长度为树的深度。

解题思路

深度优先搜索遍历

- 回溯法+动态规划，先序遍历，一个ret保存当前最优值，一个tmp保存当前值
- 递归，其实递归也是用栈实现的。（递归也太TM简洁了）

代码

代码1

```
import java.util.Stack;
public class Solution {
    public int TreeDepth(TreeNode root) {
        if (root == null)
            return 0;
        Stack<TreeNode> stack = new Stack<>();
        int ret = 0;
        int tmp = 1;
        while (!stack.isEmpty() || root != null) {
            while (root != null) {
                stack.push(root);
                root = root.left;
                tmp++;
            }
            root = stack.pop();
            tmp--;
            ret = Math.max(ret, tmp);
            root = root.right;
            if (root != null)
                tmp++;
        }
        return ret;
    }
}
```

代码2

```
public class Solution {
    public int TreeDepth(TreeNode root) {
        if (root == null)
            return 0;
        return Math.max((1 + TreeDepth(root.left)), (1 + TreeDepth(root.right)));
    }
}
```