



链滴

平衡二叉树

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518920656884>

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，判断该二叉树是否是平衡二叉树。

解题思路

左右子树的深度差距小于等于一，且左右子树也是平衡二叉树。

- 先判断整棵树，再判断子树。越底层的结点计算次数越多；
- 先判断子树再判断整棵树，返回子树的深度，标志位放在函数体外。

代码

代码1

```
public class Solution {  
    public boolean IsBalanced_Solution(TreeNode root) {  
        if (root == null)  
            return true;  
        int left = TreeDepth(root.left);  
        int right = TreeDepth(root.right);  
        if (Math.abs(left-right) <= 1)  
            return IsBalanced_Solution(root.left) && IsBalanced_Solution(root.right);  
        else  
            return false;  
    }  
  
    private int TreeDepth(TreeNode root) {  
        if (root == null)  
            return 0;  
        int left = TreeDepth(root.left)+1;  
        int right = TreeDepth(root.right)+1;  
        return Math.max((1 + TreeDepth(root.left)), (1 + TreeDepth(root.right)));  
    }  
}
```

代码2

```
public class Solution {  
    boolean ret = true;  
    public boolean IsBalanced_Solution(TreeNode root) {  
        TreeDepth(root);  
        return ret;  
    }  
  
    private int TreeDepth(TreeNode root) {  
        if (root == null)  
            return 0;  
        int left = TreeDepth(root.left)+1;  
        int right = TreeDepth(root.right)+1;
```

```
    if (Math.abs(left - right) > 1)
        ret = false;
    return Math.max(left, right);
}
}
```