[每日 LeetCode] 110. Balanced Binary Tree

作者：Hanseltu
原文链接：https://ld246.com/article/1559369807233
来源网站：链滴

原文链接 [每日LeetCode] 110. Balanced Binary Tree

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as:

a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

Example 1:

Given the following tree [3,9,20,null,null,15,7] :

```
  3
 / \
9  20
  /  \
 15   7
```

Return **true**.

Example 2:

Given the following tree[1,2,2,3,3,null,null,4,4]:

```
      1
     / \
    2   2
   / \
  3   3
 / \
4   4
```

Return **false**.

思路：本题要求判断一棵树是否是平衡二叉树。使用到了辅助函数[每日LeetCode] 104. Maximum Depth of Binary Tree，对原树进行递归依次判断左右子树的最大深度，小于1则返回true。

C++代码

```
/**
 * 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:
    bool isBalanced(TreeNode* root) {
```

```
        if (!root)
            return true;
        int lheight = maxDepth(root->left);
        int rheight = maxDepth(root->right);

        if (abs(lheight - rheight) <= 1)
            return isBalanced(root->left) && isBalanced(root->right);
        else
            return false;
    }


    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：8ms

运行内存：16.5M