

## 平衡二叉树

输入一棵二叉树的根节点，判断该树是不是平衡二叉树。如果某二叉树中任意节点的左右子树的深度相差不超过 1，那么它就是一棵平衡二叉树。

示例 1:

给定二叉树 [3,9,20,null,null,15,7]

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

返回 true 。

示例 2:

给定二叉树 [1,2,2,3,3,null,null,4,4]

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

返回 false 。

```
/**
 * 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 getdfs(TreeNode* root)  
    {  
        if(root==nullptr) return 0;  
        int leftdepth=getdfs(root->left);  
        int rightdepth=getdfs(root->right);  
        int depth=max(leftdepth,rightdepth)+1;  
        return depth;  
    }  
  
    bool isBalanced(TreeNode* root) {  
        if(root==nullptr) return true;  
        if(abs(getdfs(root->left)-getdfs(root->right))>1) return false;  
        return isBalanced(root->left)&&isBalanced(root->right);  
    }  
};
```