

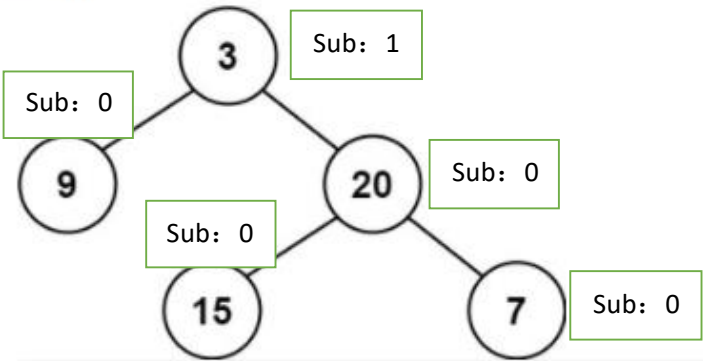
# 平衡二叉树

给定一个二叉树，判断它是否是高度平衡的二叉树。

本题中，一棵高度平衡二叉树定义为：

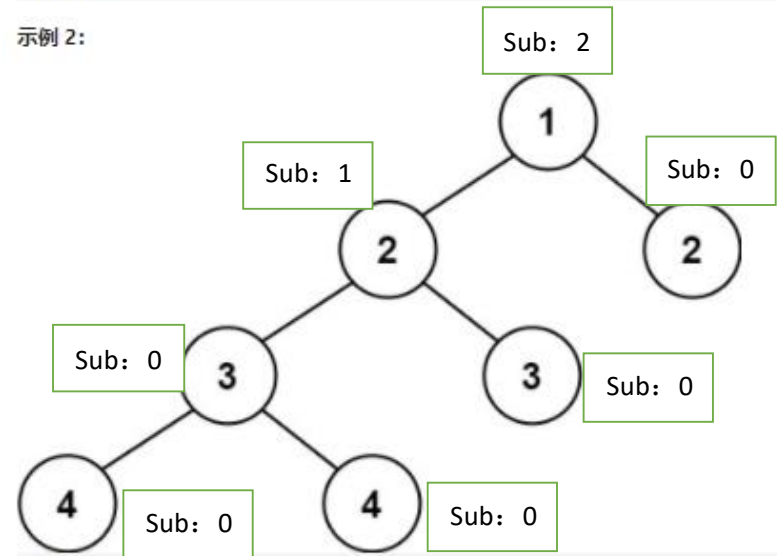
一个二叉树 每个节点 的左右两个子树的高度差的绝对值不超过 1 。

示例 1:



输入: root = [3,9,20,null,null,15,7]  
输出: true

示例 2:



输入: root = [1,2,2,3,3,null,null,4,4]  
输出: false

示例 3:

输入: root = []  
输出: true

提示:

解题思路:

题目要求: 每个节点的左右两个子树的高度差的绝对值不超过 1, 对此需要递归遍历每一个节点, 求其二叉树左右深度, 求差值作为条件判断

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */

//注意: 一个二叉树每个节点 的左右两个子树的高度差的绝对值不超过 1 。
int BinaryTreeHigh(struct TreeNode* root)
{
    if (root == NULL)
        return 0;
    int l = BinaryTreeHigh(root->left);
    int r = BinaryTreeHigh(root->right);
    return l > r? l + 1 : r + 1;
}

bool isBalanced(struct TreeNode* root){
    if(root==NULL)
        return true;
    return (abs(BinaryTreeHigh(root->left)-
BinaryTreeHigh(root->right))<=1)&&isBalanced(root->left)&&isBalanced(root->
right);
}
```