110. Balanced Binary Tree

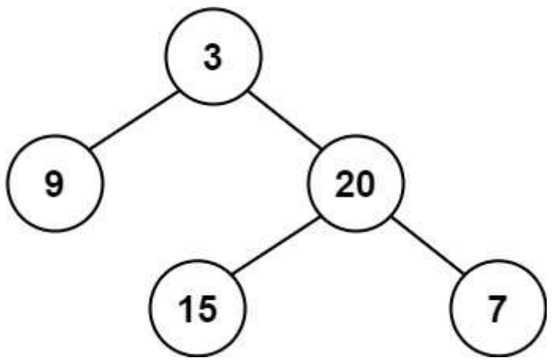
Easy 5329 289 Add to List Share

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 left and right subtrees of *every* node differ in height by no more than 1.

Example 1:




```
Input: root = [3,9,20,null,null,15,7]
Output: true
```

Example 2:

Java Autocomplete

```java
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {

    private int hightOfTree(TreeNode root){

        if(root == null)
            return 0;

        int leftSubtreeHeight = hightOfTree(root.left);
        int rightSubTreeHeight = hightOfTree(root.right);

        return 1 + Math.max(leftSubtreeHeight, rightSubTreeHeight);
    }

    public boolean isBalanced(TreeNode root) {

        if(root == null)
            return true;

        int leftHeight = hightOfTree(root.left);
        int rightHeight = hightOfTree(root.right);

        //We are checking for every single node for questions constraint "a binary tree in which the left and right subtrees of every node differ in height by no more than 1."
        return Math.abs(leftHeight - rightHeight) <= 1 && isBalanced(root.left) && isBalanced(root.right);
    }
}
```