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:

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.

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 {
    public boolean isBalanced(TreeNode root) {
        // 1:20pm - 1:33pm
        return height(root) != -1;
    }

    private int height(TreeNode root) {
        if (root == null) {
            return 0;
        }
        int leftHeight = height(root.left);
        if (leftHeight == -1) {
            return -1;
        }
        int rightHeight = height(root.right);
        if (rightHeight == -1) {
            return -1;
        }
        if (Math.abs(leftHeight - rightHeight) > 1) {
            return -1;
        }
        return 1 + Math.max(leftHeight, rightHeight);
```