

Description Solution Discuss (999+) Submissions

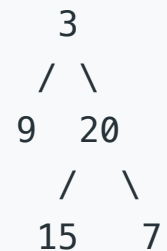
Given a binary tree, determine if it is height-balanced.

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]:



Return true.

Example 2:

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



Return false.

Java Autocomplete

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10    *         this.val = val;
11    *         this.left = left;
12    *         this.right = right;
13    *     }
14    * }
15    */
16    class Solution {
17    public boolean isBalanced(TreeNode root) {
18        // 1:20pm - 1:33pm
19        return height(root) != -1;
20    }
21
22    private int height(TreeNode root) {
23    if (root == null) {
24        return 0;
25    }
26    int leftHeight = height(root.left);
27    if (leftHeight == -1) {
28        return -1;
29    }
30    int rightHeight = height(root.right);
31    if (rightHeight == -1) {
32        return -1;
33    }
34    if (Math.abs(leftHeight - rightHeight) > 1) {
35        return -1;
36    }
37    return 1 + Math.max(leftHeight, rightHeight);
}
```

Your previous code was restored from your local storage. [Reset to default](#)