Leetcode Problem 1. (Easy)

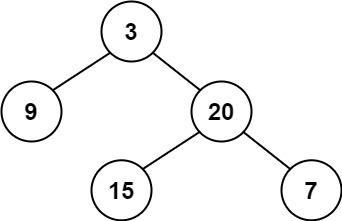
Balanced Binary Tree

Given a binary tree, determine if it is

**height-balanced**

.

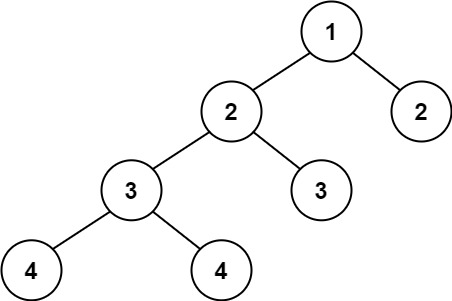
**Example 1:**



**Input:** root = [3,9,20,null,null,15,7]

**Output:** true

**Example 2:**



**Input:** root = [1,2,2,3,3,null,null,4,4]

**Output:** false

**Example 3:**

**Input:** root = []

**Output:** true

**Constraints:**

* The number of nodes in the tree is in the range [0, 5000].
* -104 <= Node.val <= 104

Link: <https://leetcode.com/problems/balanced-binary-tree/>

class TreeNode {

int val;

TreeNode left;

TreeNode right;

TreeNode(int x) { val = x; }

}

class Solution {

public boolean isBalanced(TreeNode root) {

if (root == null) {

return true;

}

int leftHeight = height(root.left);

int rightHeight = height(root.right);

if (Math.abs(leftHeight - rightHeight) > 1) {

return false;

}

return isBalanced(root.left) && isBalanced(root.right);

}

private int height(TreeNode node) {

if (node == null) {

return 0;

}

int leftHeight = height(node.left);

int rightHeight = height(node.right);

return 1 + Math.max(leftHeight, rightHeight);

}

}

