**Check Balanced**

Implement a function to check if a binary tree is balanced. For the purposes of this question, a balanced tree is defined to be a tree such that the heights of the two subtrees of any node never differ by more than one.

1 int getHeight**(**TreeNode root**)** **{**

2 **if** **(**root **==** **null)** **return** 0**;** // Base case

3 **return** Math**.**max**(**getHeight**(**root**.**left**),** getHeight**(**root**.**right**))** **+** 1**;**

4 **}**

5

6 boolean isBalanced**(**TreeNode root**)** **{**

7 **if** **(**root **==** **null)** **return** **true;**// Base case

8

9 int heightDiff **=** getHeight**(**root**.**left**)** **-** getHeight**(**root**.**right**);**

10 **if** **(**Math**.**abs**(**heightDiff**)** **>** 1**)** **{**

11 **return** **false;**

12 **}** **else** **{**//Recurse

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

14 **}**

15 **}**