110 Balanced Binary Tree

2018年3月31日 17:26

Question:

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 depth of the two subtrees of *every* node never differ by 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.

来自 <https://leetcode.com/problems/balanced-binary-tree/description/>
给定一个二叉树，确定它是高度平衡的。
对于这个问题，一棵高度平衡二叉树的定义是：
一棵二叉树中每个节点的两个子树的深度相差不会超过 1。

Solution for Python3:

```python
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.val = x
#         self.left = None
#         self.right = None

class Solution:
    def isBalanced(self, root):
        """
        :type root: TreeNode
        :rtype: bool
```

```
        """
        return self.dfsHeight(root) != -1

    def dfsHeight(self, root):
        if not root:
            return 0
        LH = self.dfsHeight(root.left)
        if LH == -1:
            return -1
        RH = self.dfsHeight(root.right)
        if RH == -1:
            return -1
        if abs(LH - RH) > 1:
            return - 1
        return max(LH, RH) + 1
```

Solution for C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    bool isBalanced(TreeNode* root) {
        return dfsHeigh(root) != -1;
    }

    int dfsHeigh(TreeNode* root) {
        if (root == NULL) {
            return 0;
        }
        int LH = dfsHeigh(root->left);
        if (LH == -1) {
            return -1;
        }
        int RH = dfsHeigh(root->right);
        if (RH == -1) {
            return -1;
        }
        if (abs(LH - RH) > 1) {
            return -1;
        }
        return max(LH, RH) + 1;
```

```
32         }
33     };
```

Appendix:

Analyse:

基于DFS。在DFS递归过程中返回当前节点的高度。当当前节点的子节点是平衡的时候，dfsHeight()函数返回一个非负的高度值，否则返回-1。根据左子树和右子树的高度值，当前节点能检查出子树是否平衡并返回相应值。