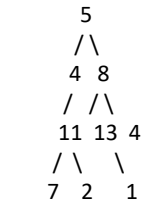
112 Path Sum

2018年3月31日 20:41

Question:

Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the given sum.
For example:
Given the below binary tree and sum = 22,



return true, as there exist a root-to-leaf path 5->4->11->2 which sum is 22.

来自 <https://leetcode.com/problems/path-sum/description/>
给定一棵二叉树和一个总和，确定该树中是否存在根到叶的路径，这条路径的所有值相加等于给定的总和。

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

# Recursive Version:
class Solution1:
    def hasPathSum(self, root, sum):
        """
        :type root: TreeNode
        :type sum: int
        :rtype: bool
        """
        if not root:
            return False
        if not root.left and not root.right:
            return root.val == sum
        return self.hasPathSum(root.left, sum - root.val) or self.hasPathSum(root.right, sum - root.val)

# Iterative Version:
class Solution2:
    def hasPathSum(self, root, sum):
        """
        :type root: TreeNode
        :type sum: int
        :rtype: bool
        """
        from collections import deque
        nodeDeq = deque([root])
        sumDeq = deque([sum])
        while nodeDeq and root:
            val = sumDeq.popleft()
            node = nodeDeq.popleft()
            if not node.left and not node.right and node.val == val:
                return True
            if node.left:
                nodeDeq.append(node.left)
                sumDeq.append(val - node.val)
            if node.right:
                nodeDeq.append(node.right)
                sumDeq.append(val - node.val)
        return False
```

Solution for C++:

```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
// Recursive Version:
class Solution1 {
public:
    bool hasPathSum(TreeNode* root, int sum) {
        if (!root) {
            return false;
        }
        if (!root->left && !root->right) {
            return sum == root->val;
        }
        return hasPathSum(root->left, sum - root->val) || hasPathSum(root->right, sum - root->val);
    }
};

// Iterative Version:
class Solution2 {
public:
    bool hasPathSum(TreeNode* root, int sum) {
        queue<TreeNode*> nodeQue;
        queue<int> sumQue;
        nodeQue.push(root);
        sumQue.push(sum);
        while (!nodeQue.empty() && root) {
            int val = sumQue.front();
            sumQue.pop();
            TreeNode* node = nodeQue.front();
            nodeQue.pop();
            if (!node->left && !node->right && node->val == val) {
                return true;
            }
            if (node->left) {
                nodeQue.push(node->left);
                sumQue.push(val - node->val);
            }
            if (node->right) {
                nodeQue.push(node->right);
                sumQue.push(val - node->val);
            }
        }
        return false;
    }
};
```