226 Invert Binary Tree

2018年4月5日 16:00

Question:

Invert a binary tree.

```
   4
  / \
 2   7
/\  /\
1  3 6  9
```

to

```
   4
  / \
 7   2
/\  /\
9  6 3  1
```

来自 <https://leetcode.com/problems/invert-binary-tree/description/>

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 Solution1:
    def invertTree(self, root):
        """
        :type root: TreeNode
        :rtype: TreeNode
        """
        if not root:
            return root
        T = self.invertTree(root.right)
        root.right = self.invertTree(root.left)
        root.left = T
        return root

class Solution2:
    def invertTree(self, root):
        """
        :type root: TreeNode
        :rtype: TreeNode
        """
        if not root:
            return None
        from collections import deque
        deq = deque([root])
        while deq:
            node = deq.popleft()
            t = node.left
            node.left = node.right
            node.right = t
            if node.left:
                deq.append(node.left)
            if node.right:
                deq.append(node.right)
        return root
```

Solution for C++:

```cpp
class Solution1 {
public:
    TreeNode* invertTree(TreeNode* root) {
        if (!root) {
            return root;
        }
        TreeNode* T = invertTree(root->right);
        root->right = invertTree(root->left);
        root->left = T;
        return root;
    }
};

class Solution2 {
public:
    TreeNode* invertTree(TreeNode* root) {
        if (!root) {
            return root;
        }
        queue<TreeNode*> que;
        que.push(root);
        while (!que.empty()) {
            TreeNode* node = que.front();
            que.pop();
            TreeNode* t = node->left;
            node->left = node->right;
            node->right = t;
            if (node->left) {
                que.push(node->left);
            }
            if (node->right) {
                que.push(node->right);
            }
        }
        return root;
    }
};
```