5月25日日报

本日学习内容

1. 学习了ui中的标签并自己运行程序
2. 完成周报
3. 每日算法题

今日算法题

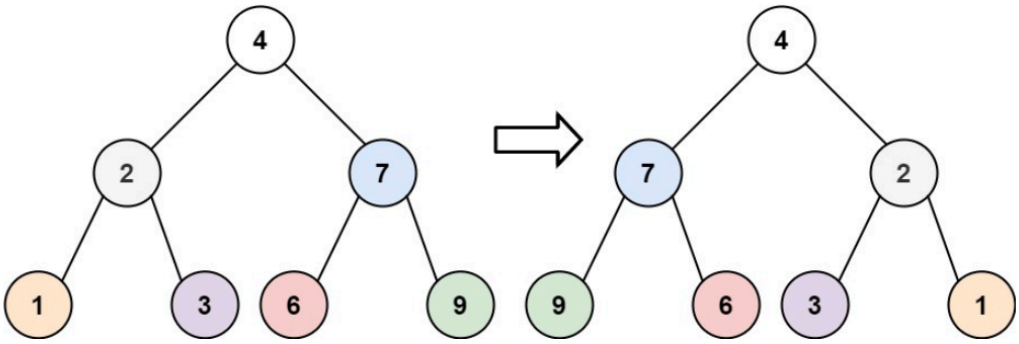
题目1：翻转二叉树

226. 翻转二叉树 已解答

简单 相关标签 相关企业 A文

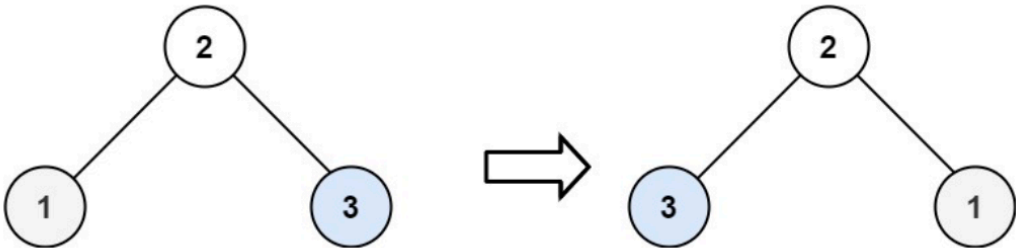
给你一棵二叉树的根节点 `root` ，翻转这棵二叉树，并返回其根节点。

示例 1：



> **输入：** root = [4,2,7,1,3,6,9]
> **输出：** [4,7,2,9,6,3,1]

示例 2：




```
class Solution {
public:
    TreeNode* invertTree(TreeNode* root) {
```

```
        queue<TreeNode *> que;
        if (root == nullptr) {
            return nullptr;
        }
        que.push(root);
        while (!que.empty()) {
            int size = que.size();
            for (int i = 0; i < size; i++) {
                TreeNode* node = que.front();
                que.pop();
                swap(node->left, node->right);
                if (node->left) {
                    que.push(node->left);
                }
                if (node->right) {
                    que.push(node->right);
                }
            }
        }
        return root;
    }
};
```

题目二：二叉树的最小深度

111. 二叉树的最小深度

已解答

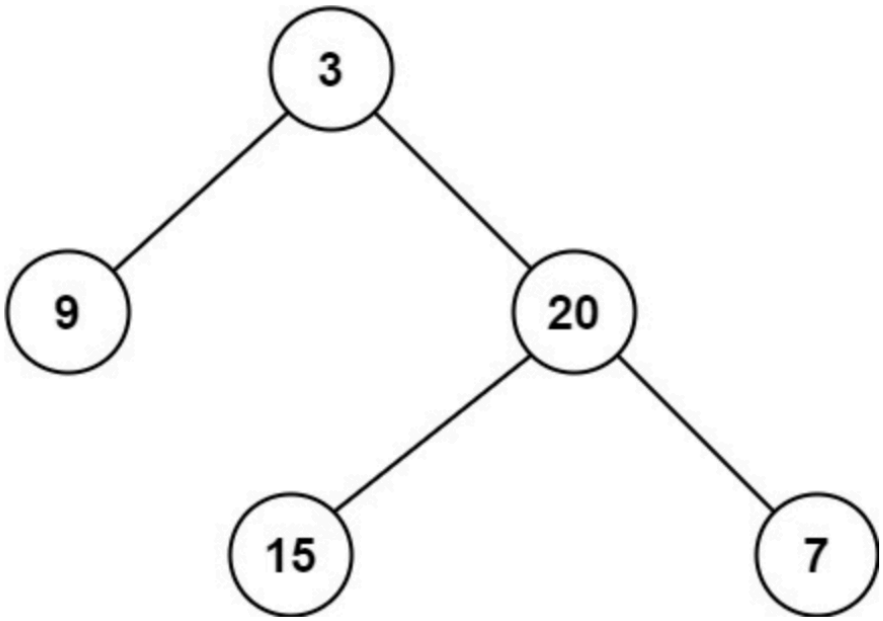
简单 相关标签 相关企业 A文

给定一个二叉树，找出其最小深度。

最小深度是从根节点到最近叶子节点的最短路径上的节点数量。

说明：叶子节点是指没有子节点的节点。

示例 1：




```
class Solution {
public:
    int minDepth(TreeNode* root) {
        if (root == nullptr) {
            return 0;
        }
        int depth = 0;
        queue<TreeNode *> que;
        que.push(root);
        while(!que.empty()) {
            int size = que.size();
            depth++;
            for (int i = 0; i < size; i++) {
                TreeNode* node = que.front();
                que.pop();
                if (node->right) {
                    que.push(node->right);
                }
```

```
                if (node->left) {
                    que.push(node->left);
                }
                if (!(node->right) && !(node->left)) {
                    return depth;
                }
            }
        }
        return depth;
    }
};
```

本日遇到的问题

1. 对->与点语法的区别不太清晰
2. 对foudation框架中一些方法只会表层使用，换种形式就不会了

明日学习计划

1. 每日算法题
2. 学习ui中的UIButton等内容