

4月24日报

本日学习内容

1. 学习二叉树的递归遍历，递归方法验证二叉搜索树，判断相同/对称/平衡二叉树
2. 学习git分支操作
3. 完成部分leetcode hot 100，总结为博客

今日算法题

题目1: [98. 验证二叉搜索树](#)

98. 验证二叉搜索树

中等

🏷 相关标签

🔒 相关企业

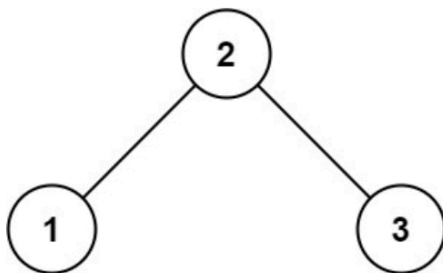
Aa

给你一个二叉树的根节点 `root`，判断其是否是一个有效的二叉搜索树。

有效 二叉搜索树定义如下：

- 节点的左子树只包含 小于 当前节点的数。
- 节点的右子树只包含 大于 当前节点的数。
- 所有左子树和右子树自身必须也是二叉搜索树。

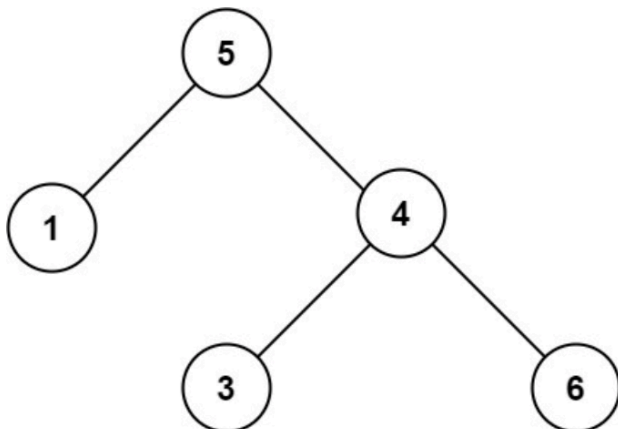
示例 1：



输入：root = [2,1,3]

输出：true

示例 2：



输入：root = [5,1,4,null,null,3,6]

输出：false

解释：根节点的值是 5，但是右子节点的值是 4。

```
class Solution {
public:
    bool isValidBST(TreeNode* root, long long left = LLONG_MIN, long long right = LLONG_MAX) {
        if (root == nullptr) {
            return true;
        }
        long long x = root->val;
        return left < x && x < right &&
            isValidBST(root->left, left, x) &&
            isValidBST(root->right, x, right);
    }
};
```

题目2: [199. 二叉树的右视图](#)

199. 二叉树的右视图

中等

🏷 相关标签

🏢 相关企业

Aa

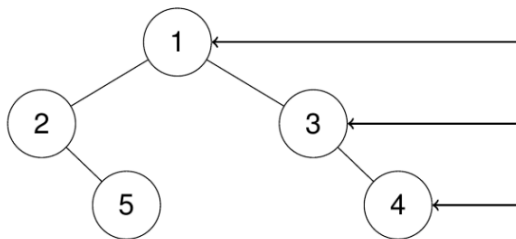
给定一个二叉树的 根节点 `root`，想象自己站在它的右侧，按照从顶部到底部的顺序，返回从右侧所能看到的节点值。

示例 1:

输入: `root = [1,2,3,null,5,null,4]`

输出: `[1,3,4]`

解释:

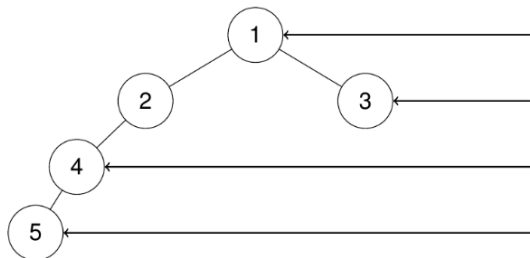


示例 2:

输入: `root = [1,2,3,4,null,null,null,5]`

输出: `[1,3,4,5]`

解释:



示例 3:

输入: `root = [1,null,3]`

输出: `[1,3]`

```
class Solution {
public:
    vector<int> rightSideView(TreeNode* root) {
        vector<int> ans;
        auto dfs = [&](this auto&& dfs, TreeNode* node, int depth) -> void {
            if (node == nullptr) {
                return;
            }
            if (depth == ans.size()) { // 这个深度首次遇到
                ans.push_back(node->val);
            }
            dfs(node->right, depth + 1); // 先递归右子树，保证首次遇到的一定是最右边的节点
        };
        dfs(root, 0);
        return ans;
    }
};
```

```
        dfs(node->left, depth + 1);  
    };  
    dfs(root, 0);  
    return ans;  
}  
};
```

明日学习计划：

1. 继续学习git操作分离head
2. 继续学习二叉树迭代遍历和层序遍历
3. 继续完成hot100数组部分，写至个人博客