4.20日报

今日学习内容：

1.代码随想录到二叉树的16。（13思路有不太会写，再研究研究）

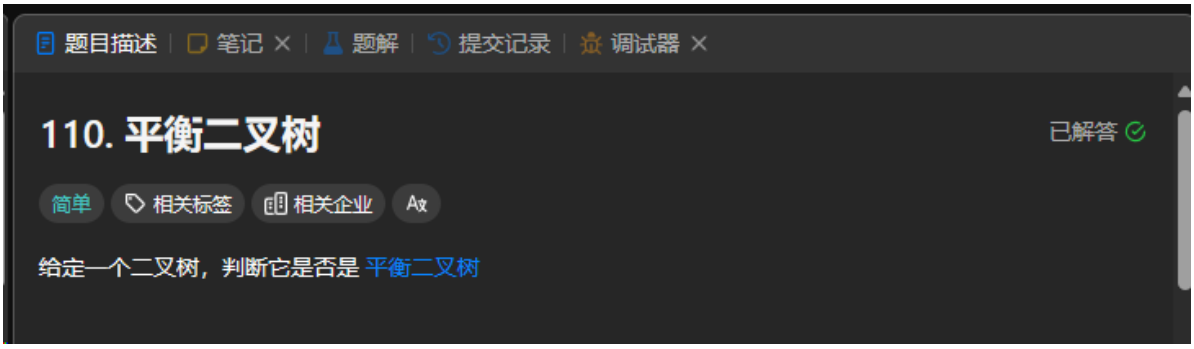
明日学习计划：

1.继续树的学习和算法题练习。

今日算法题：

原题链接：

[110. 平衡二叉树]




```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
int max(int a, int b){
    return a>b?a:b;
}

int height(struct TreeNode* root) {
    if (root == NULL) {
        return 0;
    } else {
        return max(height(root->left), height(root->right)) + 1;
    }
}

bool isBalanced(struct TreeNode* root) {
    if (root == NULL) {
        return true;
    } else {
        return fabs(height(root->left) - height(root->right)) <= 1 &&
isBalanced(root->left) && isBalanced(root->right);
    }
}
```

平衡二叉树的定义是：二叉树的每个节点的左右子树的高度差的绝对值不超过 1，所以思路就是递归的找出左右结点的最高深度作差即可。

原题链接：

[404. 左叶子之和]


```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */


int sumOfLeftLeaves(struct TreeNode* root){
    if(root==NULL){
        return 0;
    }
    int l = sumOfLeftLeaves(root->left);
    int r = sumOfLeftLeaves(root->right);

    int x = 0;
    if(root->left&&(root->left->left==NULL&&root->left->right==NULL)){
        x = root->left->val;
    }
    return l + r + x;
}
```

左叶子结点：首先其父母结点的左结点不为空，并且其左右结点为空，所以判断条件应为 if(root->left&&(root->left->left==NULL&&root->left->right==NULL))，然后递归找每一条路。总体思路是深搜。