# Assignment #8: 树为主

Updated 1704 GMT+8 Apr 8, 2025
2025 spring, Complied by 胡新璞，工学院

## 1. 题目

### LC108.将有序数组转换为二叉树

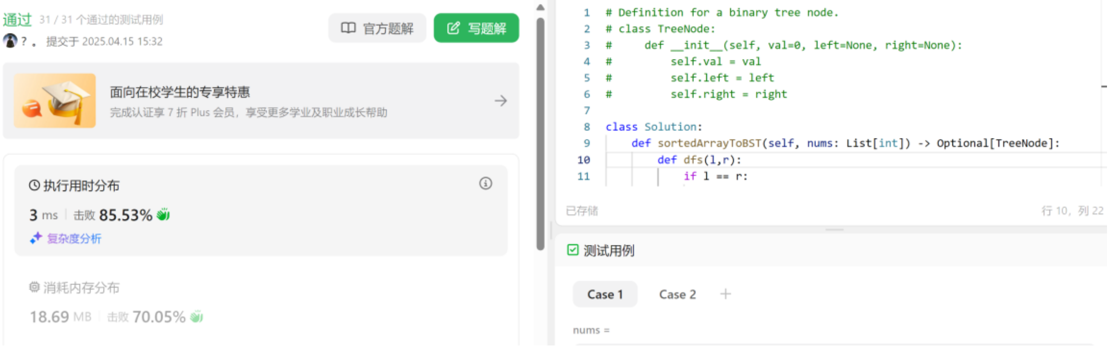
dfs, https://leetcode.cn/problems/convert-sorted-array-to-binary-search-tree/

代码：

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right

class Solution:
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        def dfs(l,r):
            if l == r:
                return None
            n = (l + r) // 2
            return TreeNode(nums[n],dfs(l,n),dfs(n + 1,r))
        return dfs(0,len(nums))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



通过 31 / 31 个通过的测试用例

? 。 提交于 2025.04.15 15:32

面向在校学生的专享特惠
完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助

执行用时分布
3 ms | 击败 85.53%

消耗内存分布
18.69 MB | 击败 70.05%

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right

class Solution:
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        def dfs(l,r):
            if l == r:
```

### M27928:遍历树

adjacency list, dfs, http://cs101.openjudge.cn/practice/27928/

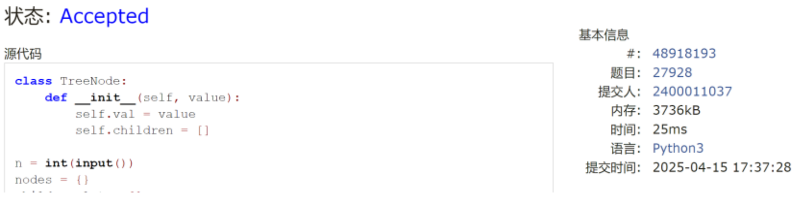
思路：

用字典。有基本正确的思路但自己独立写不出来，参考了题解

代码：

```python
class TreeNode:
    def __init__(self, value):
        self.val = value
        self.children = []
n = int(input())
nodes = {}
children_lst = []
for _ in range(n):
    lst = list(map(int, input().split()))
    nodes[lst[0]] = TreeNode(lst[0])
    for i in range(1, len(lst)):
        nodes[lst[0]].children.append(lst[i])
        children_lst.append(lst[i])
def printTree(root,nodes):
    if not root.children:
        print(root.val)
        return
    nodes_dic = {}
    nodes_dic[root.val] = root
    for child in root.children:
        nodes_dic[child] = nodes[child]
    nodes_dic_key = sorted(nodes_dic.keys())
    for node in nodes_dic_key:
        if node in root.children:
            printTree(nodes_dic[node],nodes)
        else:
            print(root.val)
val_lst = []
for value in nodes.keys():
    if not value in children_lst:
        val_lst.append(value)
printTree(nodes[val_lst[0]],nodes)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态：Accepted

源代码

```python
class TreeNode:
    def __init__(self, value):
        self.val = value
        self.children = []

n = int(input())
nodes = {}
```

基本信息

#：48918193

题目：27928

提交人：2400011037

内存：3736kB

时间：25ms

语言：Python3

提交时间：2025-04-15 17:37:28

### LC129.求根节点到叶节点数字之和

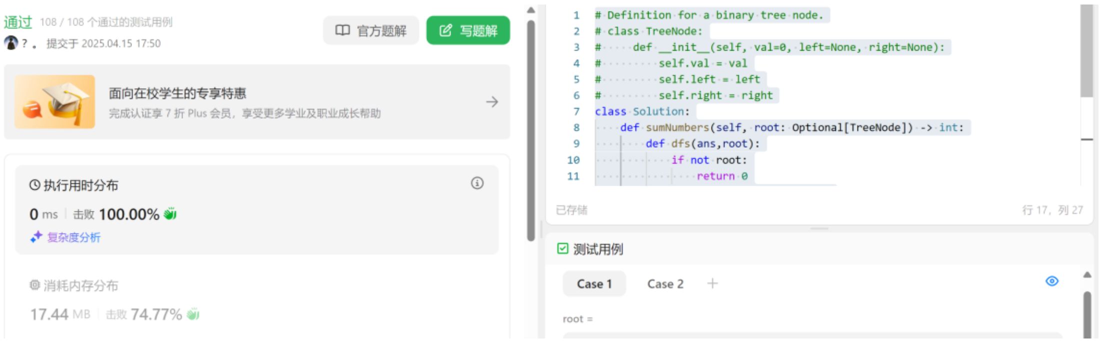
dfs, https://leetcode.cn/problems/sum-root-to-leaf-numbers/

思路：感觉是一个很常规的 dfs，遍历+找叶子节点。

代码：

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def sumNumbers(self, root: Optional[TreeNode]) -> int:
        def dfs(ans,root):
            if not root:
                return 0
            ans = ans * 10 + root.val
            if not root.left and not root.right:
                return ans
            else:
                return dfs(ans, root.left) + dfs(ans, root.right)
        return dfs(0,root)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



通过 108 / 108 个通过的测试用例

? 。 提交于 2025.04.15 17:50

执行用时分布

0 ms | 击败 100.00%

消耗内存分布

17.44 MB | 击败 74.77%

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def sumNumbers(self, root: Optional[TreeNode]) -> int:
        def dfs(ans,root):
            if not root:
                return 0
```

已存储　　行 17，列 27

☑ 测试用例

root =

### M22158:根据二叉树前中序序列建树

tree, http://cs101.openjudge.cn/practice/22158/

思路:

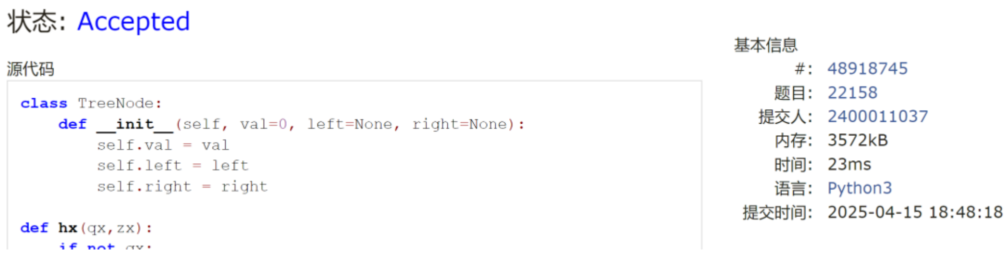
代码:

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def hx(qx,zx):
    if not qx:
        return ''
    root = qx[0]
    left_zx,right_zx = zx[0:zx.index(root)],zx[zx.index(root)+1:]
    left_qx,right_qx = qx[1:1 + len(left_zx)],qx[1 + len(left_zx):]
    left_hx,right_hx = hx(left_qx, left_zx),hx(right_qx, right_zx)
    return left_hx + right_hx + root

while True:
    try:
        qianxu = input()
        houxu = input()
        print(hx(qianxu, houxu))
    except EOFError:
        break
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

def hx(qx,zx):
    if not qx:
```

基本信息

#: 48918745

题目: 22158

提交人: 2400011037

内存: 3572kB

时间: 23ms

语言: Python3

提交时间: 2025-04-15 18:48:18

### M24729:括号嵌套树

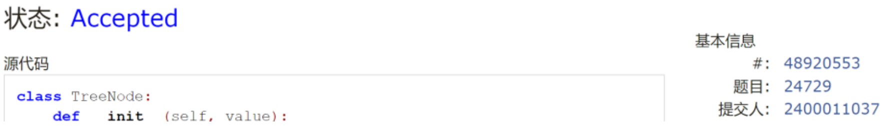
dfs, stack, http://cs101.openjudge.cn/practice/24729/

思路：中间 parse_tree 自己写总是有 bug，问了 DeepSeek。

代码：

```
class TreeNode:
    def __init__(self, value):
        self.val = value
        self.children = []
def parse_tree(s,i):
    node = TreeNode(s[i])
    i += 1
    if i < len(s) and s[i] == '(':
        i += 1
        while i < len(s) and s[i] != ')':
            child, i = parse_tree(s, i)
            node.children.append(child)
            if i < len(s) and s[i] == ',':
                i += 1
        if i < len(s) and s[i] == ')':
            i += 1
    return node, i
def qx(node,ans):
    if not node:
        return
    ans.append(node.val)
    for child in node.children:
        qx(child,ans)
def hx(node,ans):
    if not node:
        return
    for child in node.children:
        hx(child,ans)
    ans.append(node.val)
tree_list = list((input()))
root,yigebuzhidaoyoushenmeyongdedongxi = parse_tree(tree_list,0)
qianxu,houxu = [],[]
qx(root,qianxu)
hx(root,houxu)
print("".join(qianxu))
print("".join(houxu))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



状态: Accepted

源代码

```
class TreeNode:
    def   init  (self, value):
```

基本信息

#: 48920553

题目: 24729

提交人: 2400011037

### LC3510.移除最小数对使数组有序 II

doubly-linked list + heap, https://leetcode.cn/problems/minimum-pair-removal-to-sort-array-ii/

思路：

代码：

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

## 2. 学习总结和收获

<mark>如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。</mark>

对树的了解和递归解决此类题目的熟练程度有一定提升，但遇到较复杂的需要借助各种手段综合实现的问题仍然有些手忙脚乱，暂时还需要利用ai做出说明和帮助debug，比如与stack结合，不过这其中也学习到了很多经典的想法和写法。继续努力。