

# Assignment #8: 树为主

Updated 1704 GMT+8 Apr 8, 2025

2025 spring, Compiled by 郑涵予 物理学院

## 说明:

### 1. 解题与记录:

对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora <https://typoraio.cn> 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。

2. **提交安排:** 提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。

3. **延迟提交:** 如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

## 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/>

思路:

因为原数组已经有序，每次只要找到中间节点然后递归即可（用时约5min）

代码:

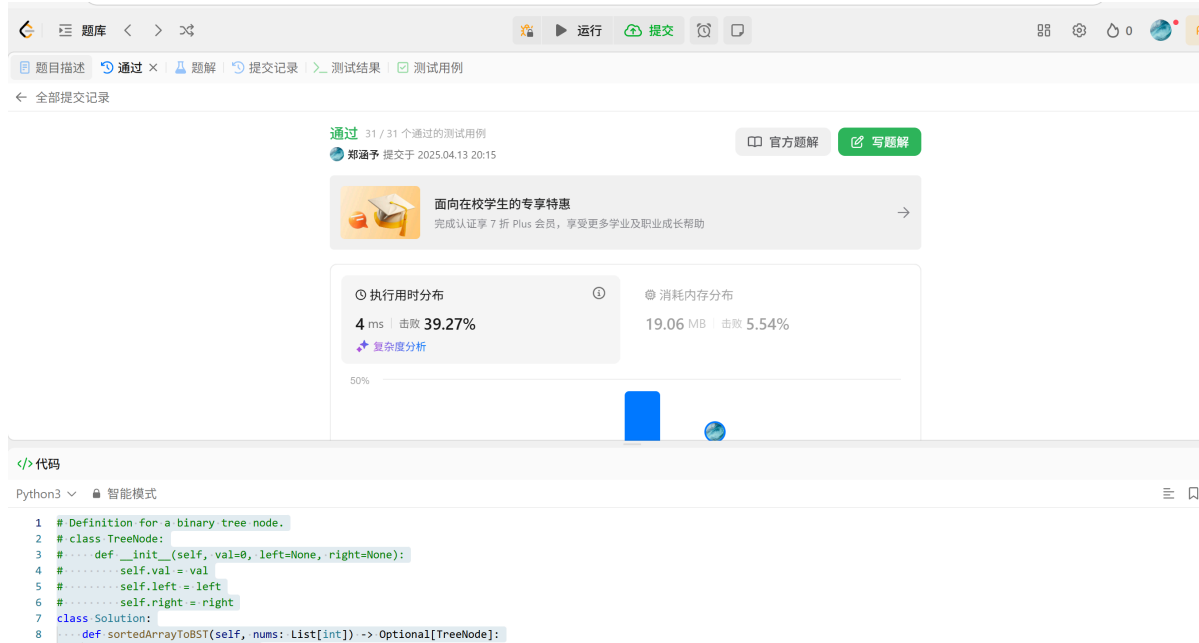
```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        def create(left, right):
```

```

        if left>right:return None
        mid=(left+right)//2
        root=TreeNode(nums[mid])
        root.left=create(left,mid-1)
        root.right=create(mid+1,right)
        return root
    return create(0,len(nums)-1)

```

代码运行截图 (至少包含有"Accepted")



## M27928:遍历树

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

思路:

按照题意直接历遍 (用时约10min)

代码:

```

s=set()
def dfs(root):
    if len(dic[root])==0 or root in s:
        print(root)
        return
    s.add(root)
    for x in dic[root]:
        dfs(x)
n=int(input())
dic={}

```

```

visited=set()
for _ in range(n):
    a=list(map(int,input().split()))
    dic[a[0]]=sorted(a)
    for i in range(1,len(a)):
        visited.add(a[i])
for key in dic.keys():
    if key not in visited:
        dfs(key)
        break

```

代码运行截图 (至少包含有"Accepted")

#48903306提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

s=set()
def dfs(root):
    if len(dic[root])==0 or root in s:
        print(root)
        return
    s.add(root)
    for x in dic[root]:
        dfs(x)
n=int(input())
dic={}
visited=set()
for _ in range(n):
    a=list(map(int,input().split()))
    dic[a[0]]=sorted(a)
    for i in range(1,len(a)):
        visited.add(a[i])
for key in dic.keys():
    if key not in visited:
        dfs(key)
        break

```

基本信息

#: 48903306  
 题目: 27928  
 提交人: 24n2400011325  
 内存: 3748kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2025-04-13 20:17:27

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 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:

直接进行dfs即可 (用时约5min)

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:

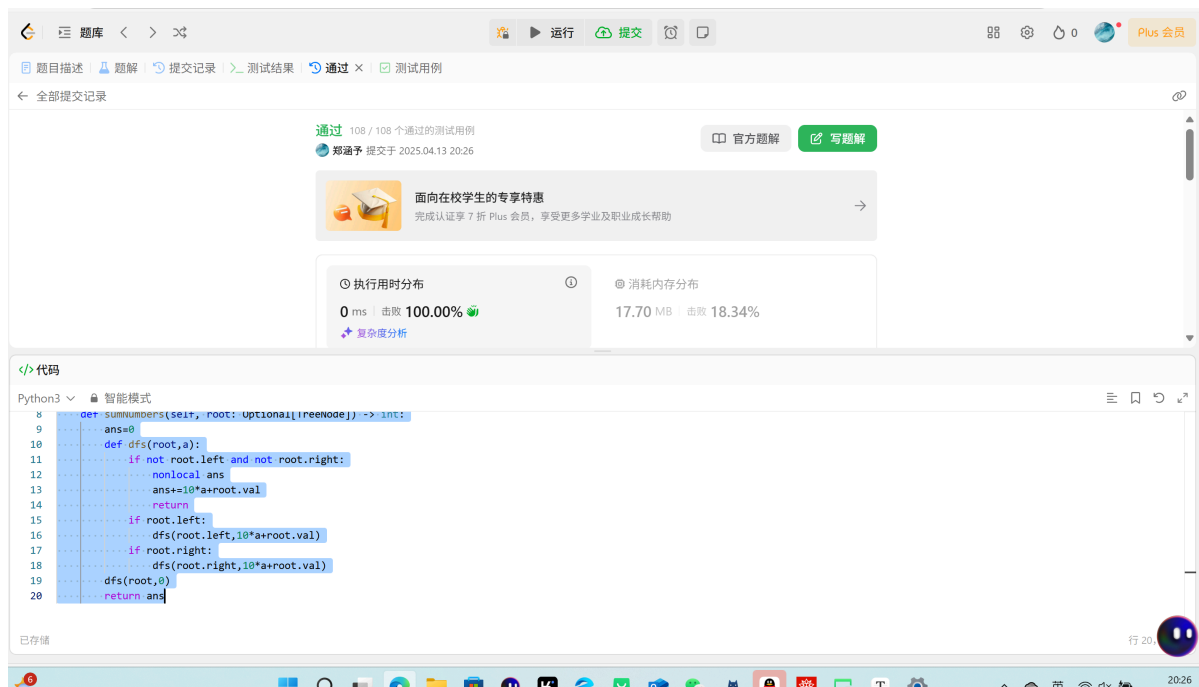
```

```

ans=0
def dfs(root,a):
    if not root.left and not root.right:
        nonlocal ans
        ans+=10*a+root.val
        return
    if root.left:
        dfs(root.left,10*a+root.val)
    if root.right:
        dfs(root.right,10*a+root.val)
dfs(root,0)
return ans

```

代码运行截图 (至少包含有"Accepted")



## M22158:根据二叉树前中序序列建树

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

思路:

经典题目，可以直接递归（用时约10min）

代码:

```

class Node():
    def __init__(self, val=""):
        self.val=val
        self.left=None

```

```

        self.right=None

def create(prel,prer,inl,inr):
    if prer<prel:
        return None
    node=Node(preorder[prel])
    k=inl
    while k<=inr:
        if inorder[k]==preorder[prel]:
            break
        k+=1
    numleft=k-inl
    node.left=create(prel+1,prel+numleft,inl,k-1)
    node.right=create(prel+numleft+1,prer,k+1,inr)
    return node

def postorder(root):
    if not root:
        return
    postorder(root.left)
    postorder(root.right)
    res.append(root.val)

while 1:
    try:
        preorder=input()
        inorder=input()
        n=len(preorder)
        root=create(0,n-1,0,n-1)
        res=[]
        postorder(root)
        print(''.join(res))
    except EOFError:
        break

```

代码运行截图 (至少包含有"Accepted")

#48797153提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class Node():
    def __init__(self, val=""):
        self.val=val
        self.left=None
        self.right=None

    def create(prel, prer, inl, inr):
        if prer<prel:
            return None
        node=Node(preorder[prel])
        k=inl
        while k<=inr:
            if inorder[k]==preorder[prel]:
                break
            k+=1
        numleft=k-inl
        node.left=create(prel+1,prel+numleft,inl,k-1)
        node.right=create(prel+numleft+1,prel,k+1,inr)
        return node

    def postorder(root):
        if not root:
            return
        postorder(root.left)
        postorder(root.right)
```

基本信息

#: 48797153  
题目: 22158  
提交人: 24n2400011325  
内存: 3648kB  
时间: 21ms  
语言: Python3  
提交时间: 2025-04-02 13:14:37

## M24729:括号嵌套树

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

思路:

还是有些麻烦的一道题，要利用栈来建树。感觉建树的逻辑就是自己看着样例直接瞪出来的，也不知道有没有什么更普适一点的办法（用时约15min）

代码:

```
class TreeNode:
    def __init__(self, x):
        self.val=x
        self.children=[]
    def add_child(self, child):
        self.children.append(child)
def preorder(root):
    if not root:return
    print(root.val,end="")
    for child in root.children:
        preorder(child)
def postorder(root):
    if not root:return
    for child in root.children:
        postorder(child)
    print(root.val,end="")
s=input()
stack=[]
root=TreeNode(None)
for x in s:
```

```

if x==')':
    root=stack.pop()
elif x==',':
    root=stack.pop()
    root.add_child(TreeNode(None))
    stack.append(root)
    root=root.children[-1]
elif x=="(":
    stack.append(root)
    root.add_child(TreeNode(None))
    root=root.children[-1]
else:
    root.val=x
preorder(root)
print()
postorder(root)

```

代码运行截图 (至少包含有"Accepted")

#48903457提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

class TreeNode:
    def __init__(self, x):
        self.val=x
        self.children=[]
    def add_child(self, child):
        self.children.append(child)
def preorder(root):
    if not root: return
    print(root.val,end=" ")
    for child in root.children:
        preorder(child)
def postorder(root):
    if not root: return
    for child in root.children:
        postorder(child)
    print(root.val,end=" ")
s=input()
stack=[]
root=TreeNode(None)
for x in s:
    if x==')':
        root=stack.pop()
    elif x==',':
        root=stack.pop()
        root.add_child(TreeNode(None))
        stack.append(root)

```

基本信息

#: 48903457  
 题目: 24729  
 提交人: 24n2400011325  
 内存: 3632kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2025-04-13 20:28:42

## 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:

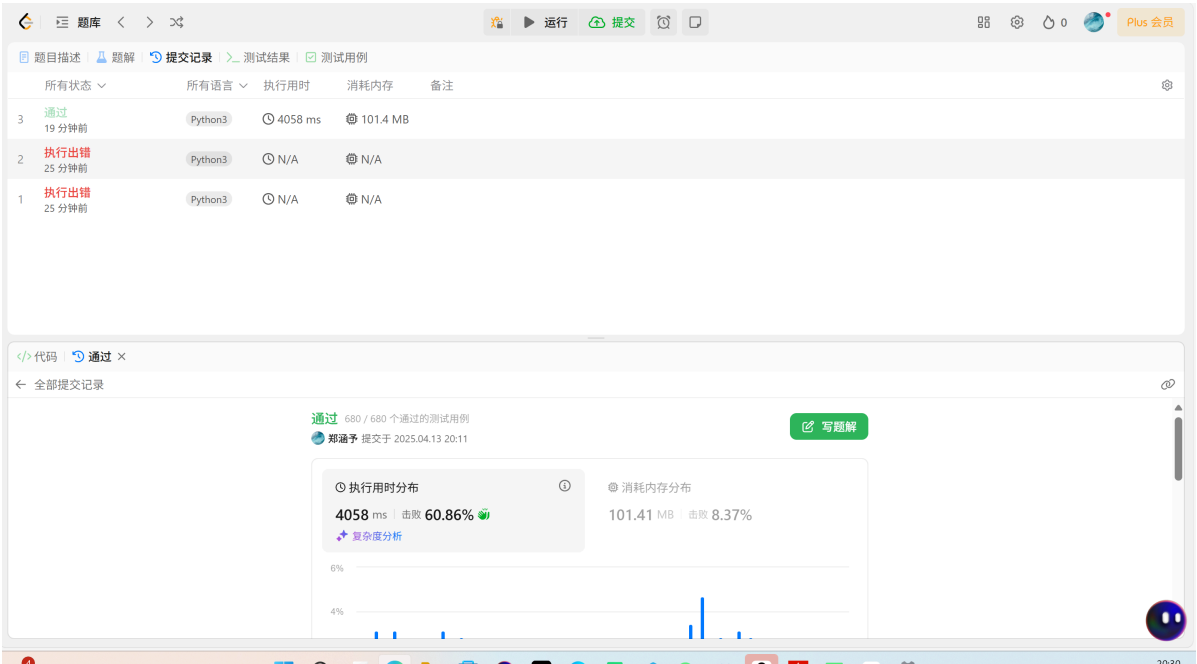
相当困难的一道题，给了tag后能够想到大致的思路，但是具体实现起来还是会遇到各种各样的问题。要是考场上真遇到这种难度的题目估计没有把握做出来（用时约1h）

代码:

```
class Solution:
    def minimumPairRemoval(self, nums: List[int]) -> int:
        n=len(nums)
        dec=0
        q=[]
        for i in range(n-1):
            if nums[i+1]<nums[i]:
                dec+=1
            q.append((nums[i]+nums[i+1],i))
        heapq.heapify(q)
        lazy=defaultdict(int)
        left=list(range(-1,n))
        right=list(range(1,n+1))
        ans=0
        while dec:
            ans+=1
            while lazy[q[0]]:
                lazy[heapq.heappop(q)]-=1
            s,i=heapq.heappop(q)
            nxt=right[i]
            if nums[i]>nums[nxt]:
                dec-=1
            pre=left[i]
            if pre>=0:
                if nums[pre]>nums[i]:
                    dec-=1
                if nums[pre]>s:
                    dec+=1
                lazy[(nums[pre]+nums[i],pre)]+=1
                heapq.heappush(q,(nums[pre]+s,pre))
            post=right[nxt]
            if post<n:
                if nums[nxt]>nums[post]:
                    dec-=1
                if s>nums[post]:
                    dec+=1
                lazy[(nums[nxt]+nums[post],nxt)]+=1
                heapq.heappush(q,(s+nums[post],i))
            nums[i]=s
            l,r=left[nxt],right[nxt]
            right[l]=r
            left[r]=l
        return ans
```



代码运行截图 (至少包含有"Accepted")



## 2. 学习总结和收获

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。

终于结束了折磨的期中，要好好开始学数算了。感觉自己做简单题和中档题还挺快的，就是做难题时还不稳定，估计还得多啃点难题。也希望期末的T不要太难。