

Assignment #8: 🎄 (2/3)

Updated 2223 GMT+8 Oct 27, 2025

2025 fall, Complied by 魏佳亮 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本人头像, 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

E108. 将有序数组转换为二叉搜索树

<https://leetcode.cn/problems/convert-sorted-array-to-binary-search-tree/>

思路: dfs递归

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 solve(l):
            if not l:
                return None
            k=len(l)//2
```

```

if not k:
    return TreeNode(l[0])
y=TreeNode(l[k])
y.left=solve(l[:k])
y.right=solve(l[k+1:])
return y
return solve(nums)

```

代码运行截图 (至少包含有"Accepted")

题目描述 | 通过 ✅ | 题解 | 提交记录

全部提交记录

通过 31 / 31 个通过的测试用例

Amazing Tereshkov... 提交于 2025.10.28 14:04

执行用时分布
2 ms 击败 88.44%

消耗内存分布
19.02 MB 击败 5.13%

智能模式

```

1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, val=0, left=None, right=None):
4 #         self.val = val
5 #         self.left = left
6 #         self.right = right
7 class Solution:
8     def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
9         def solve(l):
10             if not l:
11                 return None
12             k=len(l)//2
13             if not k:
14                 return TreeNode(l[0])
15             y=TreeNode(l[k])
16             y.left=solve(l[:k])

```

M07161: 森林的带度数层次序列存储

tree, <http://cs101.openjudge.cn/practice/07161/>

思路：按层处理

代码：

```

from collections import deque
n=int(input())
an=[ ]
for _ in range(n):
    s=input().split(' ')
    l,r=[ ],[ ]
    for i in range(len(s)):
        if i%2==0:
            l.append(s[i])
        else:
            r.append(int(s[i]))
    d={}
    s=deque([0])
    d[0]=[ ]
    k=deque([ ])
    for i in range(1,len(l)):
        d[s[0]].append(i)
        r[s[0]]-=1
        if r[s[0]]==0:
            k.append(s[0])
            s.popleft()
        if len(k)>0:
            s.append(k[0])
            k.popleft()

```

```

if not r[s[0]]:
    s.popleft()
if r[i]:
    s.append(i)
d[i]=[]
def solve(x):
    if x not in d:
        return [l[x]]
    ans=[]
    for y in d[x]:
        ans+=solve(y)
    ans+=[l[x]]
    return ans
ans=solve(0)
print(' '.join(ans))

```

代码运行截图 (至少包含有"Accepted")

#50596814提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from collections import deque
n=int(input())
ans=[]
for _ in range(n):
    s=input().split(' ')
    l,r=[],[]
    for i in range(len(s)):
        if i%2==0:
            l.append(s[i])
        else:
            r.append(int(s[i]))
    d={}
    s=deque([0])
    d[0]=[]
    k=deque([()])
    for i in range(1,len(l)):
        d[s[0]].append(i)
        r[s[0]]-=1
        if not r[s[0]]:
            s.popleft()
        if r[i]:
            s.append(i)
            d[i]=[]
    def solve(x):
        if x not in d:
            return [l[x]]
        ans=[]
        for y in d[x]:
            ans+=solve(y)
        ans+=[l[x]]
        return ans
    ans+=solve(0)
print(' '.join(ans))

```

基本信息

#: 50596814
 题目: 07161
 提交人: 24n2400011474
 内存: 3684kB
 时间: 21ms
 语言: Python3
 提交时间: 2025-10-28 12:21:28

M27928: 遍历树

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

思路：找到根节点后dfs

代码：

```
n=int(input())
d={}
for _ in range(n):
    l=list(map(int,input().split()))
    d[l[0]]=l[1:]
l=[]
for x in d.values():
    l+=x
for x in d.keys():
    if x not in l:
        r=x
        break
l=[]
def solve(x):
    t=sorted([y for y in d[x] if y<x])
    for y in t:
        solve(y)
    l.append(x)
    t=sorted([y for y in d[x] if y>x])
    for y in t:
        solve(y)
solve(r)
for x in l:
    print(x)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n=int(input())
d={}
for _ in range(n):
    l=list(map(int,input().split()))
    d[l[0]]=l[1:]
l=[]
for x in d.values():
    l+=x
for x in d.keys():
    if x not in l:
        r=x
        break
l=[]
def solve(x):
    t=sorted([y for y in d[x] if y<x])
    for y in t:
        solve(y)
    l.append(x)
    t=sorted([y for y in d[x] if y>x])
    for y in t:
        solve(y)
solve(r)
for x in l:
    print(x)

```

基本信息

#: 50592269
 题目: 27928
 提交人: 24n2400011474
 内存: 3660kB
 时间: 23ms
 语言: Python3
 提交时间: 2025-10-27 21:11:16

M129.求根节点到叶节点数字之和

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

思路: dfs

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 solve(x,s):
            nonlocal ans
            if not x.left and not x.right:
                ans+=int(s+str(x.val))
                return
            if x.left:
                solve(x.left,s+str(x.val))
            if x.right:

```

```

        solve(x.right,s+str(x.val))
    solve(root,'')
    return ans

```

代码运行截图 (至少包含有"Accepted")

```

1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, val=0, left=None, right=None):
4 #         self.val = val
5 #         self.left = left
6 #         self.right = right
7 class Solution:
8     def sumNumbers(self, root: Optional[TreeNode]) -> int:
9         ans=0
10        def solve(x,s):
11            nonlocal ans
12            if not x.left and not x.right:
13                ans+=int(s+str(x.val))
14            return
15            if x.left:
16                solve(x.left,s+str(x.val))

```

M24729: 括号嵌套树

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

思路：用stack和dict，感觉逻辑有点复杂

代码

```

s=input()
i=1
l=[0]
d={}
while i<len(s):
    if s[i]=='(':
        i+=1
        d[l[-1]]=i
        l.append(i)
        if s[i+1]==',' or s[i+1]==')':
            l.pop()
    elif s[i]==',':
        i+=1
        d[l[-1]].append(i)
        l.append(i)
        if s[i+1]==',' or s[i+1]==')':
            l.pop()
    elif s[i]==')':
        l.pop()
        i+=1
def solve(x):
    if x not in d:
        return s[x]

```

```

r=s[x]
for y in d[x]:
    r+=solve(y)
return r
print(solve(0))
def solv(x):
    if x not in d:
        return s[x]
    r=''
    for y in d[x]:
        r+=solv(y)
    r+=s[x]
    return r
print(solv(0))

```

代码运行截图 (至少包含有"Accepted")

#50591772提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

s=input()
i=1
l=[0]
d={}
while i<len(s):
    if s[i]=='(':
        i+=1
        d[l[-1]]=i
        l.append(i)
    if s[i+1]==',' or s[i+1]==')':
        l.pop()
    elif s[i]==',':
        i+=1
        d[l[-1]].append(i)
        l.append(i)
    if s[i+1]==',' or s[i+1]==')':
        l.pop()
    elif s[i]==')':
        l.pop()
        i+=1
def solve(x):
    if x not in d:
        return s[x]
    r=s[x]
    for y in d[x]:
        r+=solve(y)
    return r
print(solve(0))
def solv(x):
    if x not in d:
        return s[x]
    r=''
    for y in d[x]:
        r+=solv(y)
    r+=s[x]
    return r
print(solv(0))

```

基本信息

#: 50591772
 题目: 24729
 提交人: 24n2400011474
 内存: 3660kB
 时间: 24ms
 语言: Python3
 提交时间: 2025-10-27 20:51:09

T02775: 文件结构“图”

tree, <http://cs101.openjudge.cn/practice/02775/>

思路：用d记录层数，每个d增加一层，每个]减少一层并对f进行排序输出

代码：

```
n=1
def solve(l):
    d=0
    f=[[ ]]
    for x in l:
        if x[0]=='f':
            f[d].append(x)
        elif x[0]=='d':
            f.append([])
            d+=1
            print('|'      '*d+x)
        else:
            f[d].sort()
            for x in f[d]:
                print('|'      '*d+x)
            f.pop()
            d-=1
    f[0].sort()
    for x in f[0]:
        print(x)
while True:
    l=[ ]
    k=True
    while k:
        x=input()
        if x=='*':
            break
        if x=='#':
            k=False
            break
        l.append(x)
    if not k:
        break
    if n>1:
        print()
print(f'DATA SET {n}:')
print('ROOT')
solve(l)
n+=1
```

代码运行截图 (至少包含有"Accepted")

#50597758提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=1
def solve(l):
    d=0
    f=[[]]
    for x in l:
        if x[0]=='f':
            f[d].append(x)
        elif x[0]=='d':
            f.append([])
            d+=1
            print('|'      '*d+x)
        else:
            f[d].sort()
            for x in f[d]:
                print('|'      '*d+x)
            f.pop()
            d-=1
    f[0].sort()
    for x in f[0]:
        print(x)
while True:
    l=[]
    k=True
    while k:
        x=input()
        if x=='*':
            break
        if x=='#':
            k=False
            break
        l.append(x)
    if not k:
        break
    if n>1:
        print()
    print(f'DATA SET {n}:')
    print('ROOT')
    solve(l)
    n+=1
```

基本信息

#: 50597758
题目: 02775
提交人: 24n2400011474
内存: 3556kB
时间: 23ms
语言: Python3
提交时间: 2025-10-28 13:59:29

2. 学习总结和个人收获

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2025fall每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

每日选做还差5个题补回来，应该很快就行了，树已经跟上了，感觉确实代码比较模版，最大的难点在于如何处理输入，只要处理好了，就不会像计概那样算法错了还超时的（其实树的算法就是dfs？没啥别的算法了大概。。。）