

# Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by 李佳霖, 心理与认知科学学院

## 说明:

- 1) 请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora <https://typoraio.cn>，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。
- 2) 提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 3) 如果不能在截止前提交作业，请写明原因。

## 编程环境

（请改为同学的操作系统、编程环境等）

操作系统：macOS Sonoma 14.4

Python编程环境：VSCode

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

## 1. 题目

### 27706: 逐词倒放

<http://cs101.openjudge.cn/practice/27706/>

用时：3min

思路：用栈存储，再倒序输出即可

代码

```
#
s = input().split()
stack = []
for i in range(len(s)):
    stack.append(s[i])

res = []
for j in range(len(stack)):
    res.append(stack.pop())

print(' '.join(res[_] for _ in range(len(res))))
```

代码运行截图 (至少包含有"Accepted")

#44533932提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
s = input().split()
stack = []
for i in range(len(s)):
    stack.append(s[i])

res = []
for j in range(len(stack)):
    res.append(stack.pop())
```

基本信息

#: 44533932  
题目: 27706  
提交人: 李佳霖2000013713  
内存: 3616kB  
时间: 26ms  
语言: Python3  
提交时间: 2024-04-05 12:03:55

## 27951: 机器翻译

<http://cs101.openjudge.cn/practice/27951/>

用时: 10min

思路: 用队列存储结构, 先进先出即可

代码

```
#
from collections import deque

M, N = map(int, input().split())
words = list(map(int, input().split()))
```

```
count = 0
memory = deque()

for i in range(len(words)):

    if len(memory) <= M:

        if words[i] in memory:
            pass
        else:
            memory.append(words[i])
            count += 1
    else:
        memory.popleft()
        if words[i] in memory:
            pass
        else:
            memory.append(words[i])
            count += 1

print(count)
```

代码运行截图 (至少包含有"Accepted")

#44534017提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from collections import deque

M, N = map(int, input().split())
words = list(map(int, input().split()))
count = 0
memory = deque()

for i in range(len(words)):
```

基本信息

#: 44534017  
题目: 27951  
提交人: 李佳霖2000013713  
内存: 3648kB  
时间: 26ms  
语言: Python3  
提交时间: 2024-04-05 12:15:26

## 27932: Less or Equal

<http://cs101.openjudge.cn/practice/27932/>

思路：题目并不难，不需要自己写排序算法。但需要注意的地方是注意边界条件，例如当k=0的时候，需要找的是序列中最小的数减去1，但是这个数至少要为1。如果序列中最小的数就是1，那么没有比1更小的数满足条件，因此输出"-1"。

代码

```
#
def find_min_x(n, k, arr):
    arr.sort() # 对序列进行排序
    if k == 0:
        if arr[0] == 1:
            return "-1"
        else:
            return 1
    elif k == n:
        return arr[-1]
    else:
        if arr[k-1] == arr[k]:
            return "-1"
        else:
            return arr[k-1]

n, k = map(int, input().split())
arr = list(map(int, input().split()))

print(find_min_x(n, k, arr))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

#44535671提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
def find_min_x(n, k, arr):
    arr.sort() # 对序列进行排序
    if k == 0:
        if arr[0] == 1:
            return "-1"
        else:
            return 1
```

基本信息

#: 44535671  
题目: 27932  
提交人: 李佳霖2000013713  
内存: 10460kB  
时间: 47ms  
语言: Python3  
提交时间: 2024-04-05 15:02:57

**27948: FBI树**

<http://cs101.openjudge.cn/practice/27948/>

用时：45min

思路：按照题意建树即可，但因为有点不太熟练，花的时间比预计时间长

代码

```
#
class TreeNode():
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None
        self.parent = None
        self.seql = None
        self.seqr = None

    def buildFBI(seq, root):

        if len(seq) > 1:
            ind = len(seq)//2
            root.seql = seq[:ind]
            root.seqr = seq[ind:]
        else:
            return

        root.left = judge(root.seql)
        root.right = judge(root.seqr)
        buildFBI(root.seql, root.left)
        buildFBI(root.seqr, root.right)

    return

    def postorderTraversal(root):
        result = []
        if root:
            result.extend(postorderTraversal(root.left))
            result.extend(postorderTraversal(root.right))
            result.append(root.val)

        return result

    def judge(seq):

        if 1 in seq and 0 in seq:
            node = TreeNode('F')
        elif 1 in seq and 0 not in seq:
```

```
node = TreeNode('I')
else:
    node = TreeNode('B')

return node

N = int(input())
seq = list(map(int,input()))
root = judge(seq)

buildFBI(seq, root)
results = postorderTraversal(root)
print(''.join(results))
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

#44534406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.left = None
        self.right = None
        self.parent = None
        self.seql = None
        self.seqr = None

    def buildFBI(self, seq, root):
        if len(seq) > 1:
            ind = len(seq) // 2
            root.seql = seq[:ind]
            root.seqr = seq[ind:]
        else:
            return
```

基本信息

#: 44534406  
题目: 27948  
提交人: 李佳霖2000013713  
内存: 4200kB  
时间: 28ms  
语言: Python3  
提交时间: 2024-04-05 13:17:25

## 27925: 小组队列

<http://cs101.openjudge.cn/practice/27925/>

思路：这道题做的时候超时了（自己又臭又长的代码就不放上来了）后来看了群里大佬的代码发现原来可以把几个队伍的队列分而治之并不用一起组织，这样就节省了搜索的时间。感觉思路又受到了启发。

代码

```
#
from collections import deque

t = int(input())
teams = {i: deque(map(int, input().split())) for i in range(t)}
queue = deque()
group_queue = {i: deque() for i in range(t)} #这里比较关键，不需要放到一个队列里

while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        person = int(command[1])
        for i in range(t):
            if person in teams[i]:
                group_queue[i].append(person)
            if i not in queue:
                queue.append(i)
            break
    elif command[0] == 'DEQUEUE':
        group = queue[0]
        print(group_queue[group].popleft())
        if not group_queue[group]:
            queue.popleft()
```

代码运行截图 (AC代码截图，至少包含有"Accepted")

#44540888提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from collections import deque

t = int(input())
teams = {i: deque(map(int, input().split())) for i in range(t)}
queue = deque()
group_queue = {i: deque() for i in range(t)}
```

基本信息

#: 44540888  
 题目: 27925  
 提交人: 李佳霖2000013713  
 内存: 4228kB  
 时间: 1267ms  
 语言: Python3  
 提交时间: 2024-04-05 22:02:00

## 27928: 遍历树

<http://cs101.openjudge.cn/practice/27928/>

思路：主要难点在于遍历时候的逻辑以及如何存放无顺序输入的节点上

代码

```
#
class Node():
    def __init__(self, x):
        self.val = x
        self.parent = None
        self.children = []

    def traverse(self):
        if self.children == []:
            print(self.val)
        else:
            tmp_nodes = self.children + [self]
            tmp_nodes.sort(key=lambda x: x.val)
            #排序之后叶结点和根结点比大小，如果是第一个就直接输出，然后再看叶子结点有没有叶子结点
            for node in tmp_nodes:
                if node.val != self.val:
                    node.traverse()
                else:
                    print(node.val)

n = int(input())
nodes = {}
for i in range(n):
    info = list(map(int, input().split()))
    if info[0] not in nodes:
        nodes[info[0]] = Node(info[0])

    for j in info[1:]:
        if j not in nodes:
            nodes[j] = Node(j)
        nodes[j].parent = nodes[info[0]]
        nodes[info[0]].children.append(nodes[j])

for node in nodes:
    if nodes[node].parent is None:
        root = nodes[node]
        break

root.traverse()
```



代码运行截图 (AC代码截图, 至少包含有"Accepted")

#44540721提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
class Node():
    def __init__(self, x):
        self.val = x
        self.parent = None
        self.children = []

    def traverse(self):
        if self.children == []:
            print(self.val)
```

基本信息

#: 44540721  
题目: 27928  
提交人: 李佳霖2000013713  
内存: 3740kB  
时间: 26ms  
语言: Python3  
提交时间: 2024-04-05 21:49:52

## 2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ “2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

感觉这次上机的题目难度适中，两个小时内应该AC了4道题，比起作业则简单了很多。每道题的考点感觉只要拿捏到需要运用的数据结构之后便能够很快上手。一些模版题还是需要背的更熟一点，比如后序遍历。另外还是需要多练题看看大佬的代码启发一下思路。