Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Complied by 同学的姓名、院系

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 Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

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/

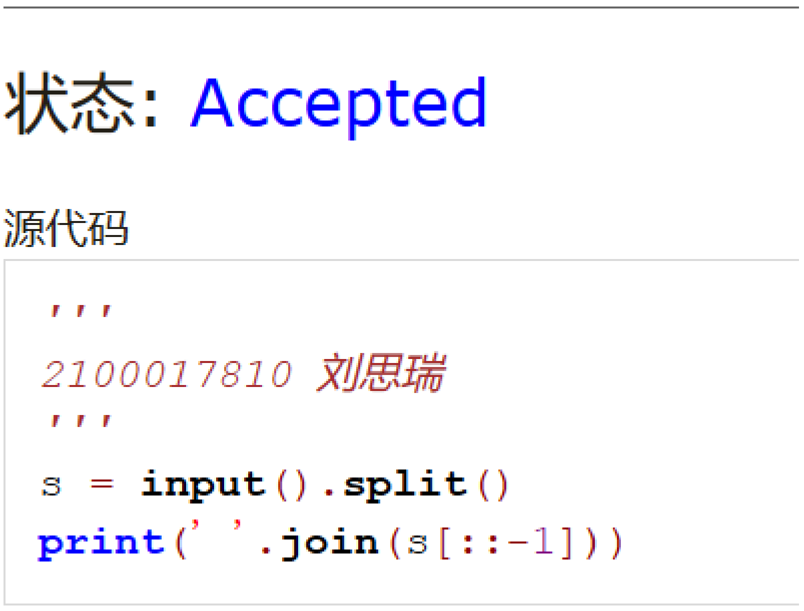
思路：

字符串切片

代码

```python
'''
2100017810 刘思瑞
'''
s = input().split()
print(' '.join(s[::-1]))
```

代码运行截图



状态: Accepted

源代码

```
'''
2100017810 刘思瑞
'''
s = input().split()
print(' '.join(s[::-1]))
```

27951: 机器翻译

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

思路:

FIFO

代码

```python
'''
2100017810 刘思瑞
'''
from collections import deque
M,N = map(int,input().split())
l = list(map(int,input().split()))
m = deque()
count = 0
for i in l:
    if i in m:
        continue
    if len(m) == M:
        m.popleft()
    m.append(i)
    count +=1
print(count)
```

代码运行截图

状态: Accepted

源代码

```
'''
2100017810 刘思瑞
'''
from collections import deque
M,N = map(int,input().split())
l = list(map(int,input().split()))
m = deque()
count = 0
for i in l:
    if i in m:
        continue
    if len(m) == M:
        m.popleft()
    m.append(i)
    count +=1
print(count)
```

27932: Less or Equal

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

思路:

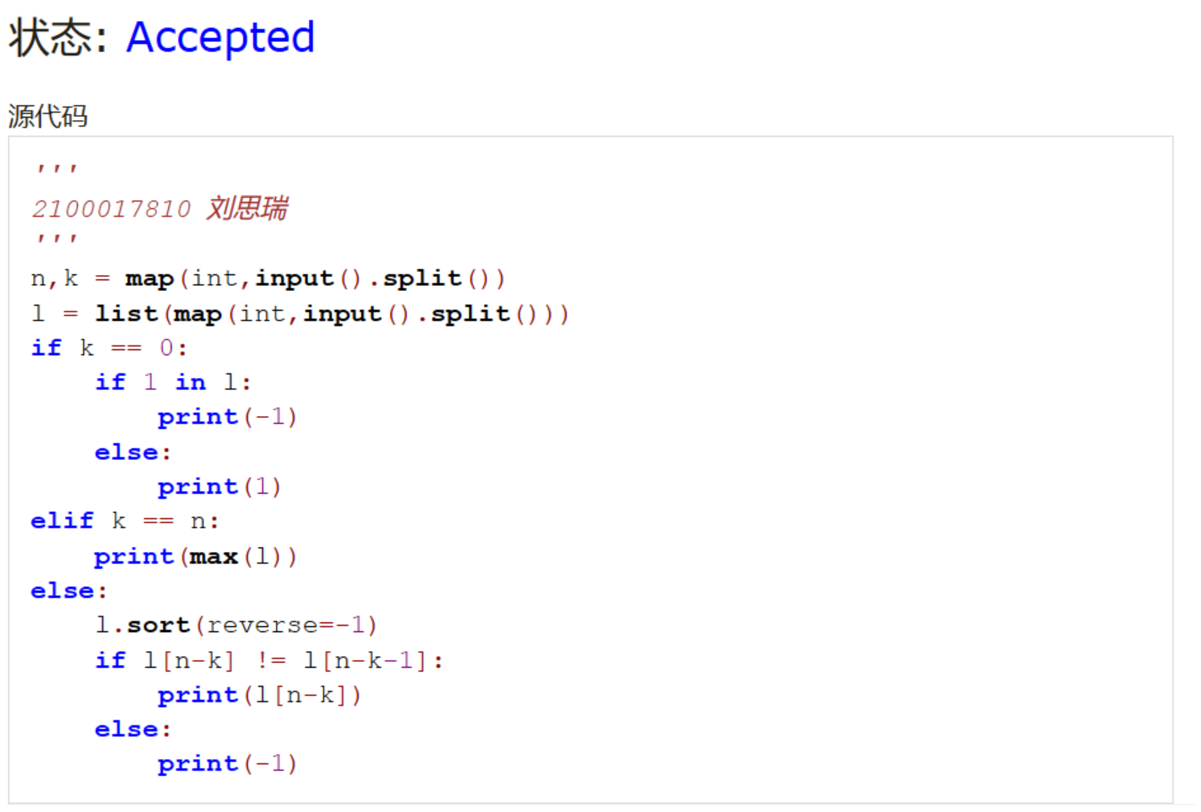
排序

代码

```
'''
2100017810 刘思瑞
'''
n,k = map(int,input().split())
l = list(map(int,input().split()))
if k == 0:
    if 1 in l:
        print(-1)
    else:
        print(1)
```

```python
elif k == n:
    print(max(l))
else:
    l.sort(reverse=-1)
    if l[n-k] != l[n-k-1]:
        print(l[n-k])
    else:
        print(-1)
```

代码运行截图



状态: Accepted

源代码

```python
'''
2100017810 刘思瑞
'''
n,k = map(int,input().split())
l = list(map(int,input().split()))
if k == 0:
    if 1 in l:
        print(-1)
    else:
        print(1)
elif k == n:
    print(max(l))
else:
    l.sort(reverse=-1)
    if l[n-k] != l[n-k-1]:
        print(l[n-k])
    else:
        print(-1)
```

27948: FBI树

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

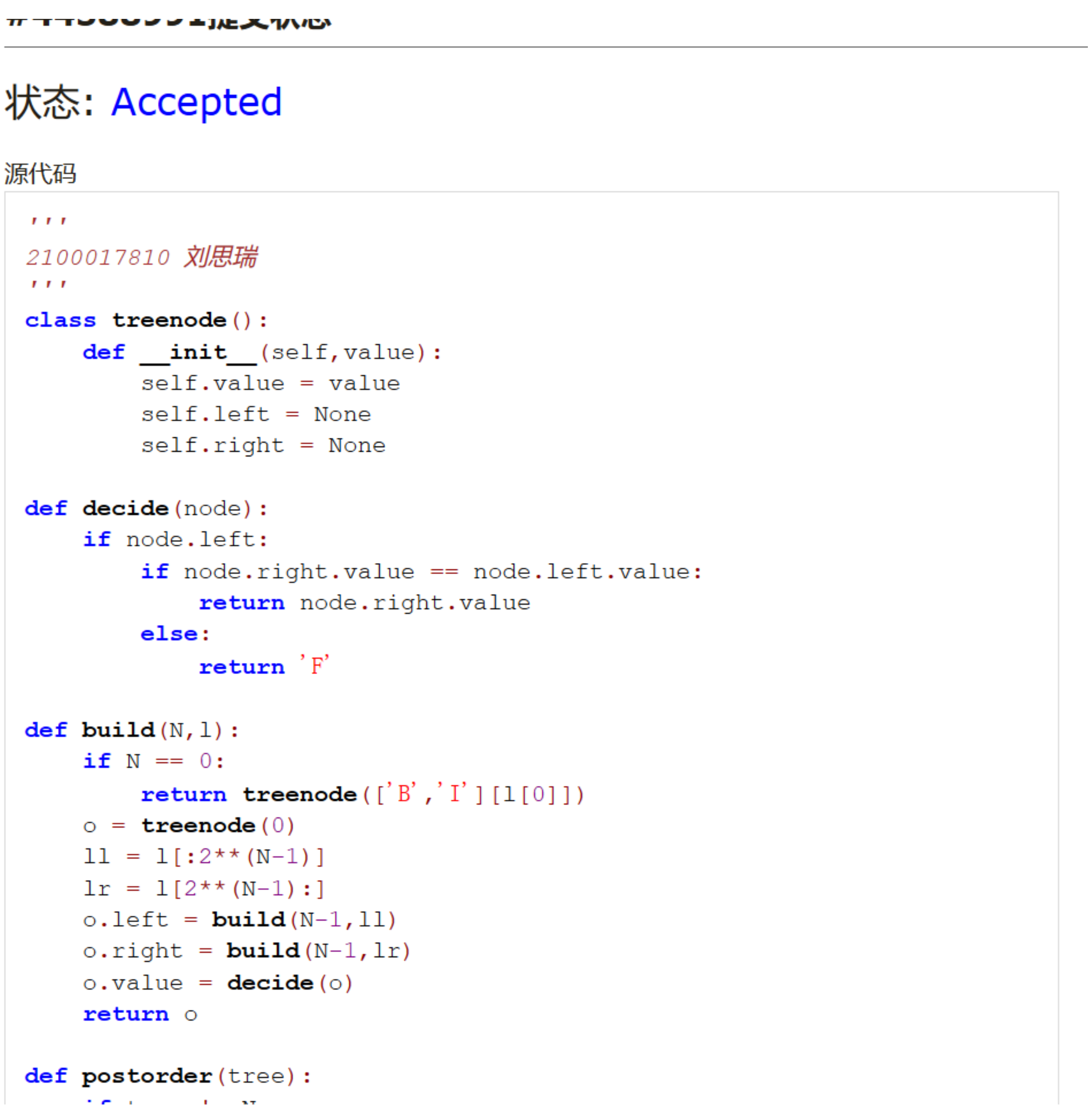
思路：

递归建树

代码

```python
'''
2100017810 刘思瑞
'''
class treenode():
    def __init__(self,value):
        self.value = value
        self.left = None
        self.right = None

def decide(node):
    if node.left:
        if node.right.value == node.left.value:
            return node.right.value
        else:
            return 'F'

def build(N,l):
    if N == 0:
        return treenode(['B','I'][l[0]])
    o = treenode(0)
    ll = l[:2**(N-1)]
    lr = l[2**(N-1):]
    o.left = build(N-1,ll)
    o.right = build(N-1,lr)
    o.value = decide(o)
    return o

def postorder(tree):
    if tree != None:
        postorder(tree.left)
        postorder(tree.right)
        print(tree.value,end='')


N = int(input())
s = list(map(int,list(input())))
postorder(build(N,s))
```

代码运行截图



状态: Accepted

源代码

```
'''
2100017810 刘思瑞
'''
class treenode():
    def __init__(self,value):
        self.value = value
        self.left = None
        self.right = None

def decide(node):
    if node.left:
        if node.right.value == node.left.value:
            return node.right.value
        else:
            return 'F'

def build(N,l):
    if N == 0:
        return treenode(['B','I'][l[0]])
    o = treenode(0)
    ll = l[:2**(N-1)]
    lr = l[2**(N-1):]
    o.left = build(N-1,ll)
    o.right = build(N-1,lr)
    o.value = decide(o)
    return o

def postorder(tree):
```

27925: 小组队列

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

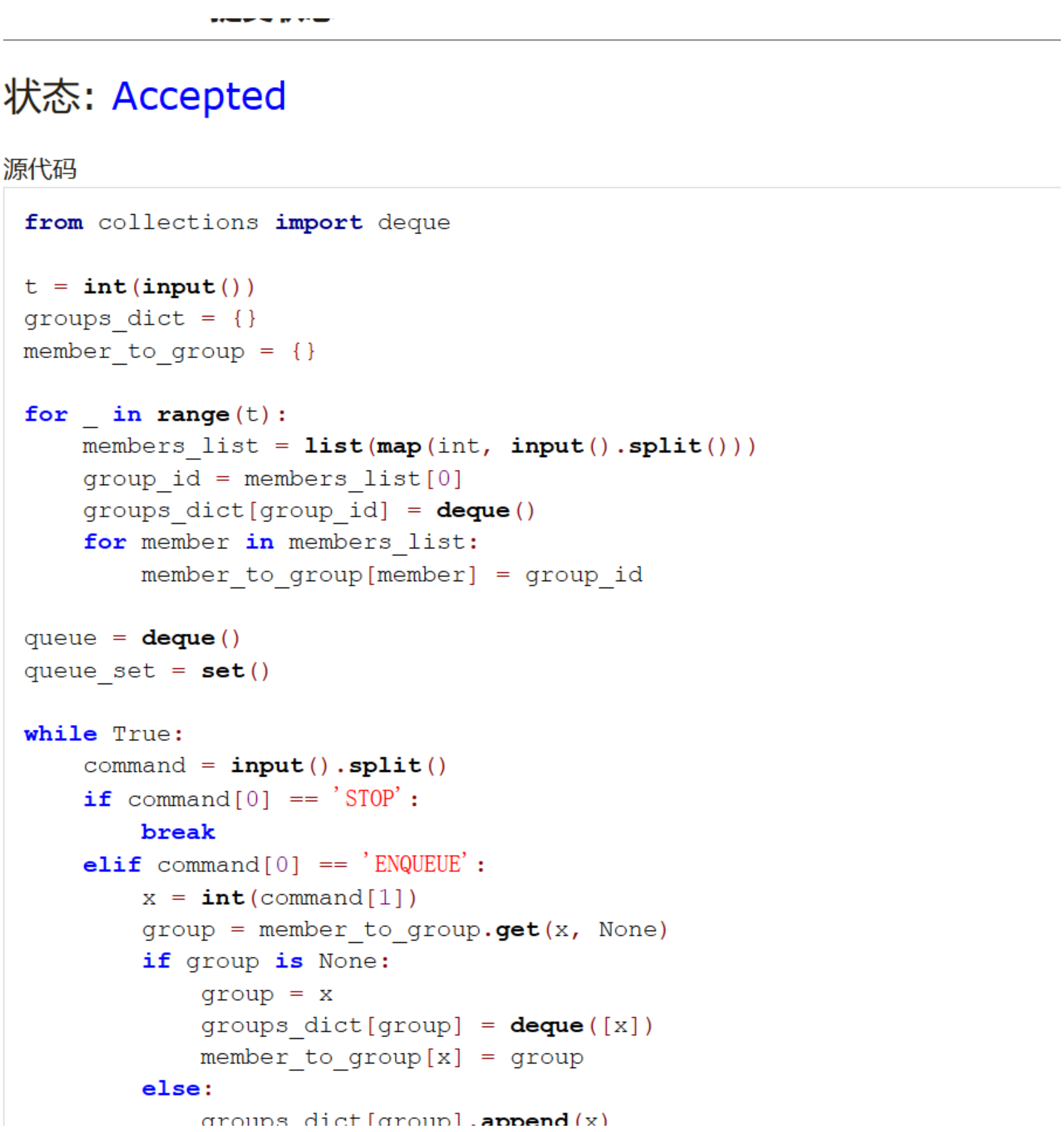
思路:

分组来考虑，记录最前面的人

代码

```python
from collections import deque

t = int(input())
groups_dict = {}
member_to_group = {}

for _ in range(t):
    members_list = list(map(int, input().split()))
    group_id = members_list[0]
    groups_dict[group_id] = deque()
    for member in members_list:
        member_to_group[member] = group_id

queue = deque()
queue_set = set()

while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        x = int(command[1])
        group = member_to_group.get(x, None)
        if group is None:
            group = x
            groups_dict[group] = deque([x])
            member_to_group[x] = group
        else:
            groups_dict[group].append(x)
        if group not in queue_set:
            queue.append(group)
            queue_set.add(group)
    elif command[0] == 'DEQUEUE':
        if queue:
            group = queue[0]
            x = groups_dict[group].popleft()
            print(x)
            if not groups_dict[group]:
                queue.popleft()
                queue_set.remove(group)
```

代码运行截图



状态: Accepted

源代码

```
from collections import deque

t = int(input())
groups_dict = {}
member_to_group = {}

for _ in range(t):
    members_list = list(map(int, input().split()))
    group_id = members_list[0]
    groups_dict[group_id] = deque()
    for member in members_list:
        member_to_group[member] = group_id

queue = deque()
queue_set = set()

while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        x = int(command[1])
        group = member_to_group.get(x, None)
        if group is None:
            group = x
            groups_dict[group] = deque([x])
            member_to_group[x] = group
        else:
            groups_dict[group].append(x)
```

27928: 遍历树

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

思路:

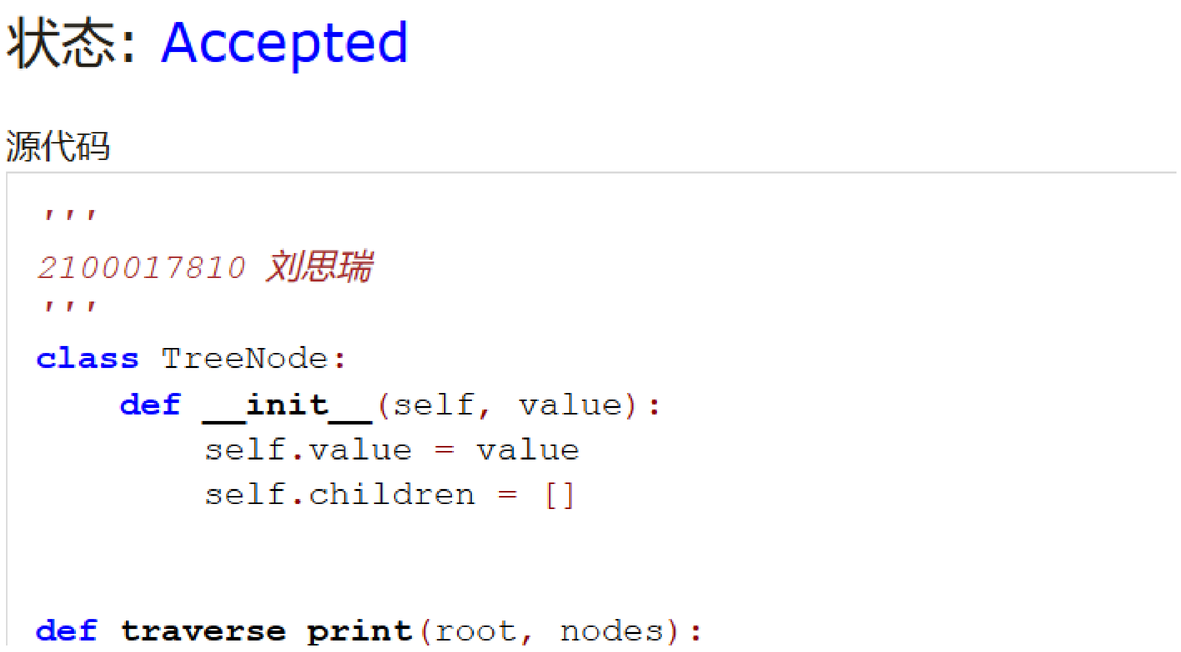
先判断父节点再递归

代码

```
'''
2100017810 刘思瑞
'''
class treenode():
    def __init__(self,value,child):
        self.value = value
        self.child = child
```

```python

def sortorder(tree):
    global hasvis,node
    while True:
        if tree != None:
            temp = []
            for i in [tree.value]+tree.child:
                if i not in hasvis:
                    temp.append(i)
            if not temp :
                return
            m = min(temp)
            print(m)
            hasvis.add(m)
            sortorder(node[m])

n = int(input())
node = dict()
for i in range(n):
    s = list(map(int,input().split()))
    v,l = s[0],s[1:]
    node[v] = treenode(v,l)
    if i == 0:
        origin  = node[v]
    if origin.value in l:
        origin = node[v]
hasvis = set()
sortorder(origin)
```

代码运行截图



状态: Accepted

源代码

```python
'''
2100017810 刘思瑞
'''
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.children = []


def traverse print(root, nodes):
```

2. 学习总结和收获

对树有了进一步的认识，并且感觉对于遍历和建树的递归操作更加熟练了