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：

代码

```
#
```

代码运行截图 ==（至少包含有"Accepted"）==

27951: 机器翻译

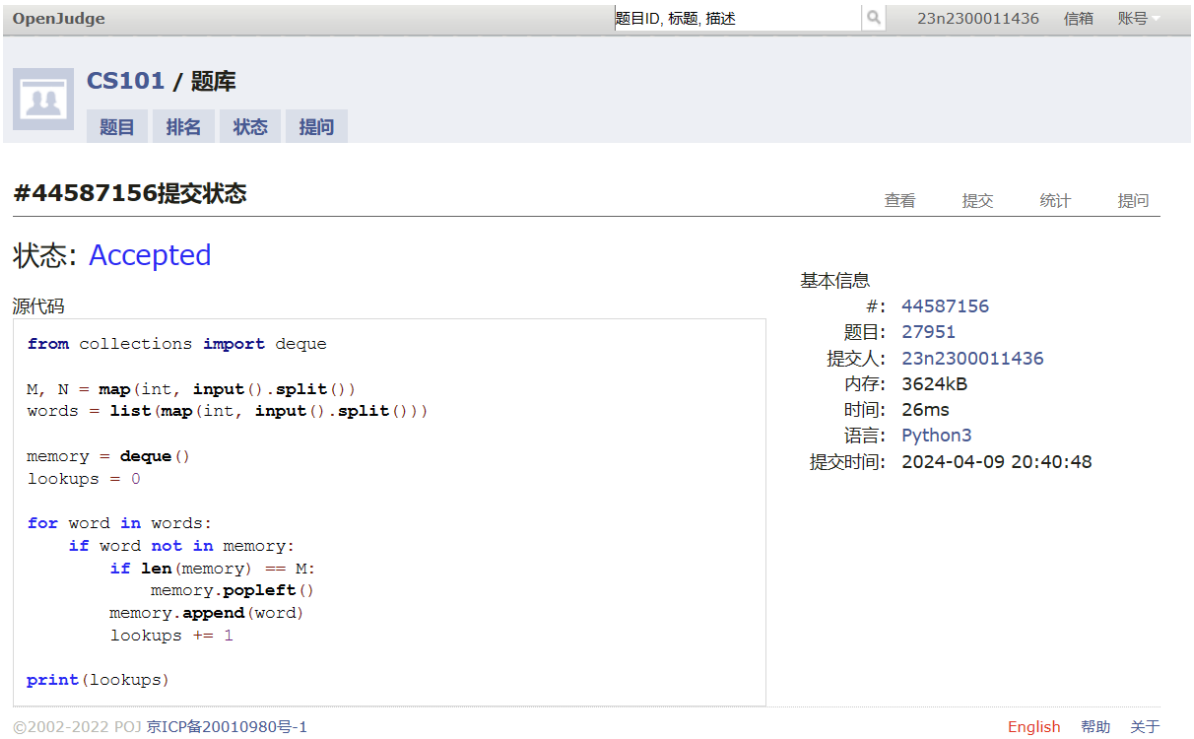
http://cs101.openjudge.cn/practice/27951/

思路:

代码

```
from collections import deque

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

memory = deque()
lookups = 0

for word in words:
    if word not in memory:
        if len(memory) == M:
            memory.popleft()
        memory.append(word)
        lookups += 1

print(lookups)
```

代码运行截图



27932: Less or Equal

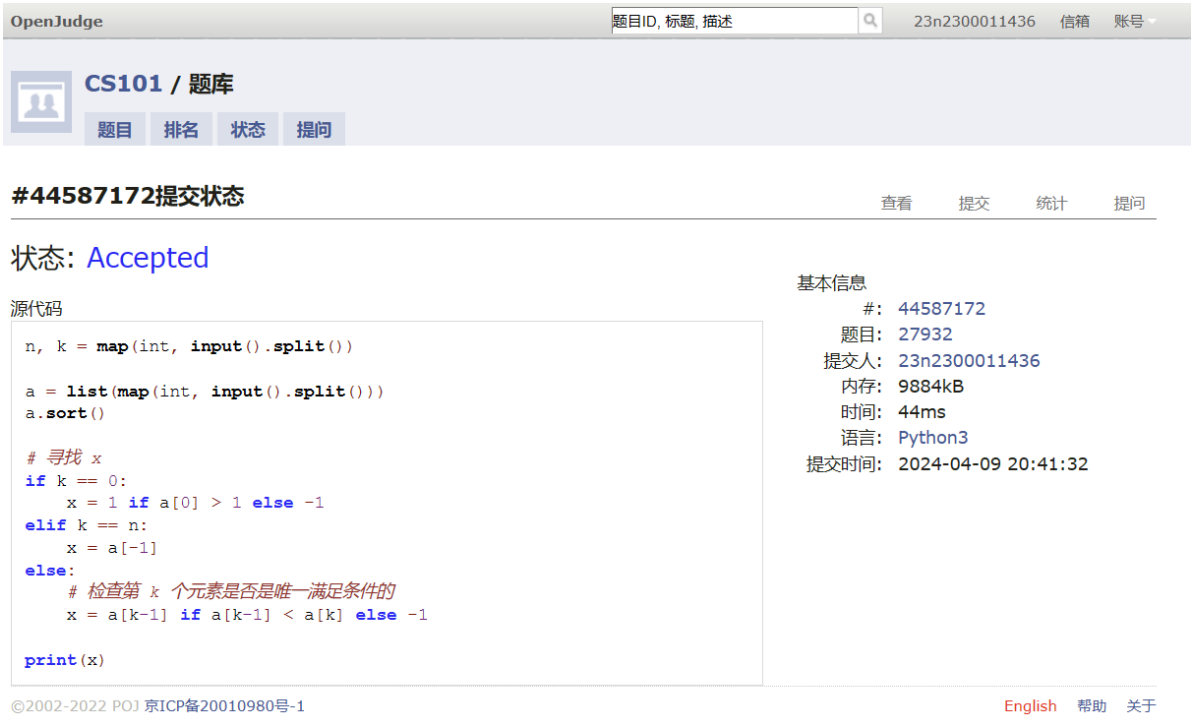
http://cs101.openjudge.cn/practice/27932/

思路:

代码

```
n, k = map(int, input().split())

a = list(map(int, input().split()))
a.sort()

# 寻找 x
if k == 0:
    x = 1 if a[0] > 1 else -1
elif k == n:
    x = a[-1]
else:
    # 检查第 k 个元素是否是唯一满足条件的
    x = a[k-1] if a[k-1] < a[k] else -1

print(x)
```

代码运行截图



27948: FBI树

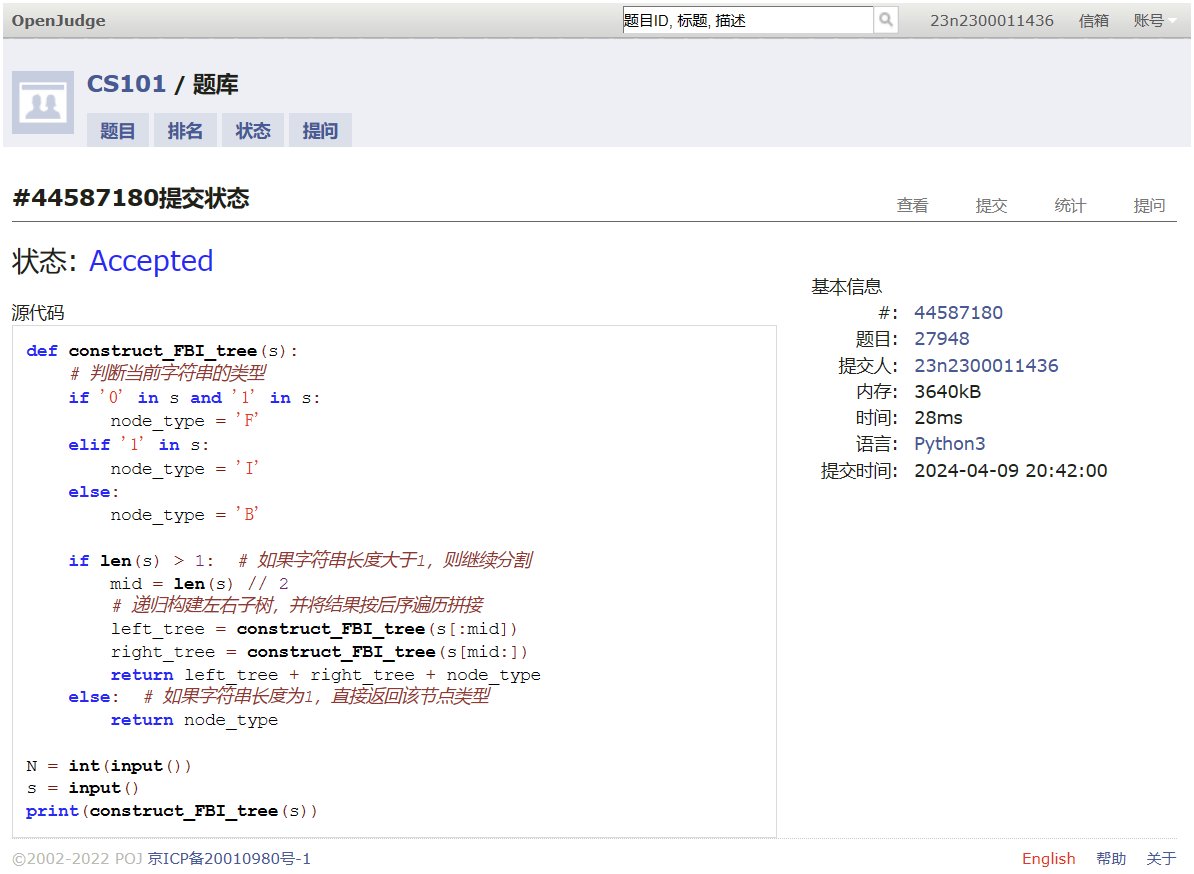
http://cs101.openjudge.cn/practice/27948/

思路:

代码

```
def construct_FBI_tree(s):
    # 判断当前字符串的类型
    if '0' in s and '1' in s:
        node_type = 'F'
    elif '1' in s:
        node_type = 'I'
    else:
        node_type = 'B'

    if len(s) > 1:  # 如果字符串长度大于1，则继续分割
        mid = len(s) // 2
        # 递归构建左右子树，并将结果按后序遍历拼接
        left_tree = construct_FBI_tree(s[:mid])
        right_tree = construct_FBI_tree(s[mid:])
        return left_tree + right_tree + node_type
    else:  # 如果字符串长度为1，直接返回该节点类型
        return node_type

N = int(input())
s = input()
print(construct_FBI_tree(s))
```

代码运行截图



27925: 小组队列

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

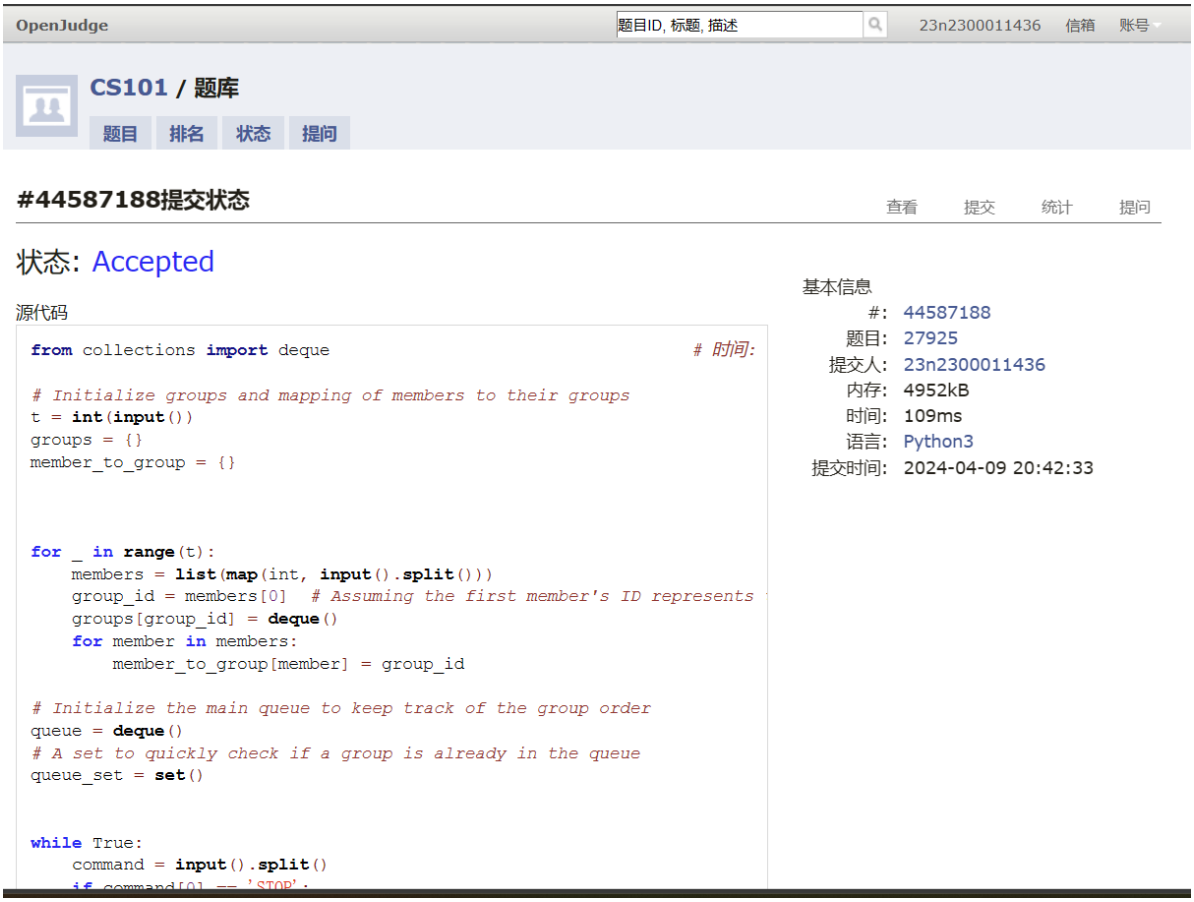
思路:

代码

```
from collections import deque                    # 时间: 105ms

# Initialize groups and mapping of members to their groups
t = int(input())
groups = {}
member_to_group = {}



for _ in range(t):
    members = list(map(int, input().split()))
    group_id = members[0]  # Assuming the first member's ID represents the group
ID
    groups[group_id] = deque()
    for member in members:
        member_to_group[member] = group_id
```

```
# Initialize the main queue to keep track of the group order
queue = deque()
# A set to quickly check if a group is already in the queue
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)
        # Create a new group if it's a new member not in the initial list
        if group is None:
            group = x
            groups[group] = deque([x])
            member_to_group[x] = group
        else:
            groups[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[group].popleft()
            print(x)
            if not groups[group]:  # If the group's queue is empty, remove it
from the main queue
                queue.popleft()
                queue_set.remove(group)
```

代码运行截图



27928: 遍历树

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

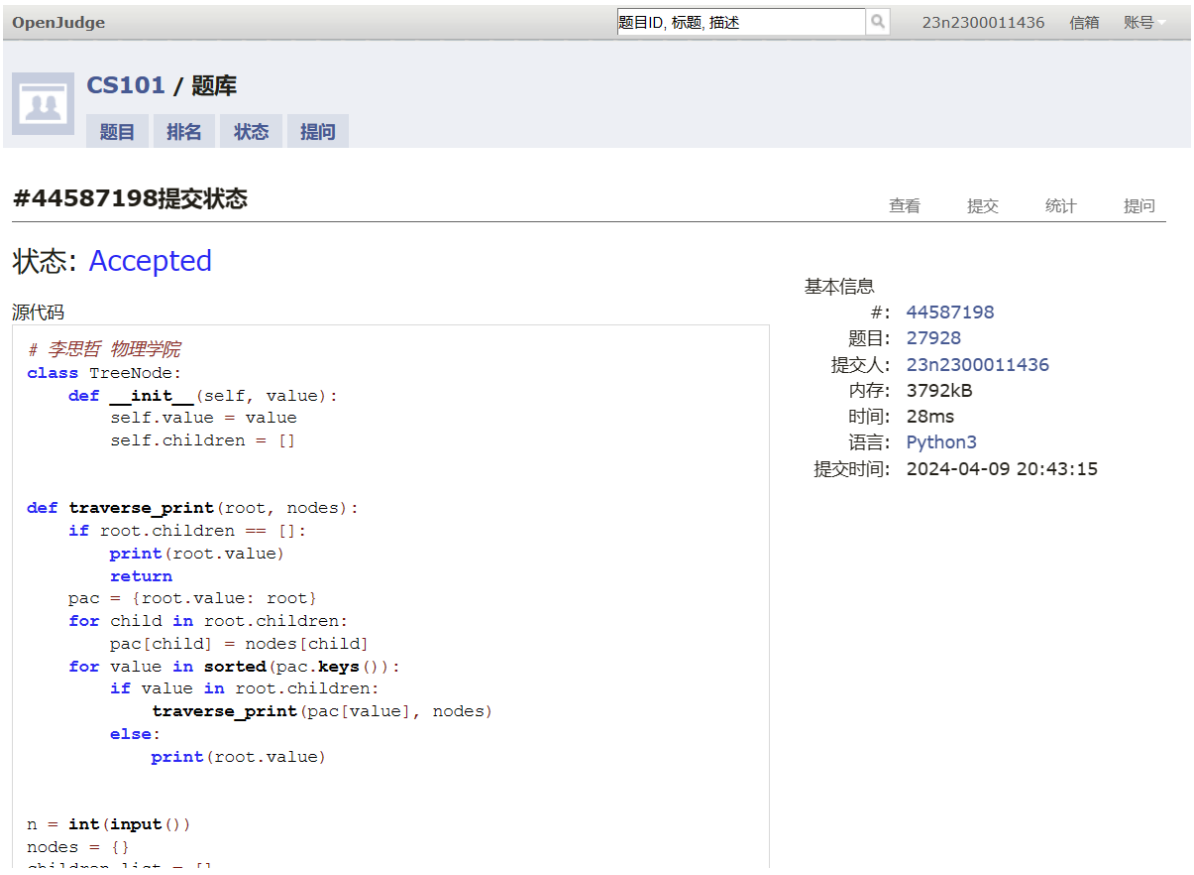
思路:

代码

```
# 李思哲 物理学院
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.children = []


def traverse_print(root, nodes):
    if root.children == []:
        print(root.value)
        return
    pac = {root.value: root}
    for child in root.children:
        pac[child] = nodes[child]
    for value in sorted(pac.keys()):
```

```
        if value in root.children:
            traverse_print(pac[value], nodes)
        else:
            print(root.value)


n = int(input())
nodes = {}
children_list = []
for i in range(n):
    info = list(map(int, input().split()))
    nodes[info[0]] = TreeNode(info[0])
    for child_value in info[1:]:
        nodes[info[0]].children.append(child_value)
        children_list.append(child_value)
root = nodes[[value for value in nodes.keys() if value not in children_list][0]]
traverse_print(root, nodes)
```

代码运行截图



2. 学习总结和收获

这周期中考完再计时做一遍