Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Complied by ==张宇帆 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 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：把输入倒过来即可

代码

```
#
S = input().split()
S.reverse()
print(' '.join(S))
```

代码运行截图 ==（至少包含有"Accepted"）==

#44515917提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
S = input().split()
S.reverse()
print(' '.join(S))
```

基本信息

#: 44515917
题目: E27706
提交人: 2200013720
内存: 3492kB
时间: 22ms
语言: Python3
提交时间: 2024-04-03 15:13:03

27951: 机器翻译

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

思路：常规的队列思路

代码

```
#
queue = []
N,M = map(int,input().split())
words = input().split()
result = 0
for word in words:
    if word not in queue:
        if len(queue) == N:
            queue.pop(0)
            queue.append(word)
        else:
            queue.append(word)
        result += 1
print(result)
```

代码运行截图 ==（至少包含有"Accepted"）==

#44516058提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
queue = []
N,M = map(int,input().split())
words = input().split()
result = 0
for word in words:
    if word not in queue:
        if len(queue) == N:
            queue.pop(0)
            queue.append(word)
        else:
            queue.append(word)
        result += 1
print(result)
```

基本信息

#: 44516058
题目: E27951
提交人: 2200013720
内存: 3628kB
时间: 24ms
语言: Python3
提交时间: 2024-04-03 15:18:12

27932: Less or Equal

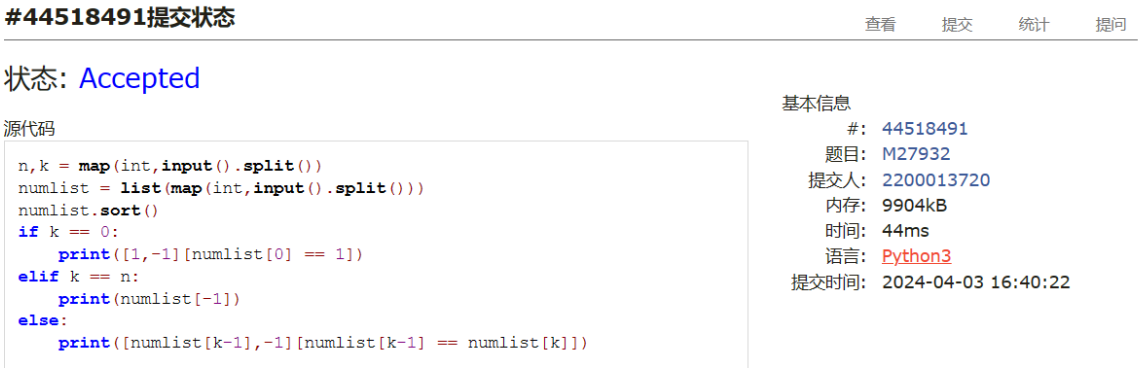
http://cs101.openjudge.cn/practice/27932/

思路：排序后判断即可。难点在于边界条件的判断，一开始k=0就直接输出0导致WA，然后就没有细想了，做到后面突然想到还有可能有多个1，所以改了一下AC了

代码

```
#
n,k = map(int,input().split())
numlist = list(map(int,input().split()))
numlist.sort()
if k == 0:
    print([1,-1][numlist[0] == 1])
elif k == n:
    print(numlist[-1])
else:
    print([numlist[k-1],-1][numlist[k-1] == numlist[k]])
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



27948: FBI树

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

思路：看到题目给的递归思路后果断放弃写树，直接上手写递归，然后就AC了，这道题就是题目给的理解好就很容易（虽然因为写得太顺于是自己想了好几个示例花了比较多的时间）

代码

```
#
def change(s):
    if len(s) == 1:
        if s == '1':
            return 'I'
        elif s == '0':
            return 'B'
```

```
    else:
        if '0' in s and '1' in s:
            root = 'F'
        else:
            root = ['B','I']['1' in s]
        idx = len(s)//2
        return change(s[:idx]) + change(s[idx:]) + root

N = int(input())
S = input()
print(change(S))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
def change(s):
    if len(s) == 1:
        if s == '1':
            return 'I'
        elif s == '0':
            return 'B'
    else:
        if '0' in s and '1' in s:
            root = 'F'
        else:
            root = ['B','I']['1' in s]
        idx = len(s)//2
        return change(s[:idx]) + change(s[idx:]) + root

N = int(input())
S = input()
print(change(S))
```

基本信息

#:	44516530
题目:	M27948
提交人:	2200013720
内存:	3636kB
时间:	25ms
语言:	Python3
提交时间:	2024-04-03 15:32:55

27925: 小组队列

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

思路:
思路很简单，依据have判断小组是否存在，存在对应的队尾位置；group存储学生对应的组号；然后每次进队看一下学生的组存不存在，存在就依据have存的位置插入，并更新排在当前小组后头的小组的队尾位置，不存在就进队尾并创建一个新队；每次出队注意更新一下每个组的队尾位置即可

代码

```
#
queue = []
group = {}
have = {}
t = int(input())
for i in range(t):
    people = input().split()
    for p in people:
        group[p] = i
order = list(map(str,input().split()))
while not order[0] == 'STOP':
```

```
    command = order[0]
    if command == 'ENQUEUE':
        person = order[1]
        code = group[person]
        if code in have:
            for i in have:
                if have[i][0] >= have[code][0]: # 更改处，之前是对所有组的队尾进行了
+1，显然是不对的
                    have[i][0] += 1
            have[code][1] += 1
            queue.insert(have[code][0], person)
        else:
            have[code] = [len(queue),1]
            queue.append(person)
    elif command == 'DEQUEUE':
        result = queue.pop(0)
        code = group[result]
        for i in have:  # 这里不用改成上面的形式，因为是队首出队
            have[i][0] -= 1
        have[code][1] -= 1
        if have[code][1] == 0:
            del have[code]
        print(result)
    order = list(map(str,input().split()))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态: Accepted

源代码

```
queue = []
group = {}
have = {}
t = int(input())
for i in range(t):
    people = input().split()
    for p in people:
        group[p] = i
order = list(map(str,input().split()))
```

基本信息

#:	44525494
题目:	27925
提交人:	2200013720
内存:	5304kB
时间:	131ms
语言:	Python3
提交时间:	2024-04-04 11:29:49

27928: 遍历树

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

思路:
思路就是先储存节点和对应索引，然后依据索引再建树，补充父节点，更新子序列为树节点，然后遍历即可

代码

```
#
class TreeNode(object):
    def __init__(self, val, child = None, parent = None):
        self.root = val
        self.child = child
```

```
        self.parent = parent

def build(trees, position):
    for t in trees:
        if not t.child == None:
            newchild = []
            for c in t.child:
                trees[position[c]].parent = t
                newchild.append(trees[position[c]])
            t.child = newchild
    return trees

def loop(treenode):
    result = []
    if treenode.child == None:
        result.append(str(treenode.root))
    else:
        idx = len(treenode.child)
        for i in range(len(treenode.child)):
            if treenode.child[i].root > treenode.root: #超过根值了，先加入根值再接着遍
历
                idx = i
                break
            else:
                result += loop(treenode.child[i])
        result.append(str(treenode.root))
        for rest in treenode.child[idx:]:
            result += loop(rest)
    return result

trees = []
position = {}
n = int(input())
for _ in range(n):
    numlist = list(map(int,input().split()))
    position[numlist[0]] = len(trees)
    trees.append(TreeNode(numlist[0], child = sorted(numlist[1:])))
trees = build(trees, position)
for t in trees:
    if t.parent == None:
        result = loop(t)
        print('\n'.join(result))
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

状态：Accepted

源代码

```
class TreeNode(object):
    def __init__(self, val, child = None, parent = None):
        self.root = val
        self.child = child
        self.parent = parent

def build(trees, position):
    for t in trees:
        if not t.child == None:
            newchild = []
```

基本信息

#：44532890
题目：27928
提交人：2200013720
内存：3792kB
时间：29ms
语言：Python3
提交时间：2024-04-05 10:29:51

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。==

考试AC5，但其实应该是AC4，两道tough题确实卡了我很久，具体情境为：

小组队列：感觉这道题最大的难点在于太绕了，一会要依据学生编码找小组，一会又要判断队列里还有没有人，一会可能还得想想会不会超内存，能考虑到的点其实一下子就能考虑到，但正是由于顾虑的点太多导致写起来束手束脚的。考试写的时候就是这种心态，最后忍不了直接多建了一个have然后写出来了，可惜虽然考试后群里同学讨论的点我考虑到了（就是出队后小组没人了），但考试提交的代码仍然是错误的，第二天改了个判断就AC了。看了群里大佬的代码也觉得很妙，不像是我能写出来（

遍历树：考试的时候下意识地以为不用parent，写到最后五分钟才屈服加入parent，但还是没做出来。很好笑的一点就是，我能够把遍历很好地写出来（下面的loop就是考试时写出来的），但！是！我没能成功建树。考试完也迟迟想不出仅依靠一个trees列表怎么建树（其实是不知道怎么用索引，用index顾虑超时）。后来索性加了个字典position就写好了

这次考试成功地打击了我的自信心（当然还有每天难度增高的每日选做），让我意识到不能吃老本了，虽然该进行的计划还在进行（学习并查集，图等），但似乎跟不上大部队的节奏了。而且这次遍历树的失误让我意识到自己对树的掌握还没那么好，所以还要多练点树的题目。第一次深刻体会到学得越多，要补的漏洞就越来越多。而其他课程给的作业压力真的太大了，虽然也是写代码但之前完全没接触过，所以其实每周花了更多的时间在上面。之前还安慰自己数算的每日选做都有做，也不算是很摆烂，但随着题目难度增高，每日选做我也逐渐做不上了。说实话越来越焦虑了，总想着用更多的时间来学习与debug。

还是给自己打打气吧，看到模型训练成功，代码能够AC还是会开心的嘛，期末不求AC6好歹努努力能够整个AC5吧。这学期都过去一半了！一切都会好起来的！