

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by 赵语涵 生命科学学院

非考试时间计时完成AC5

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

操作系统: windows11

Python编程环境: Spyder IDE 5.2.2

1. 题目

27706: 逐词倒放

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

思路: 简单地split以后倒序输出即可

代码

```
1 #赵语涵2300012254
2 result = []
3 for i in input().split():
4     result.append(i)
5 print(' '.join(result[::-1]))
```

代码运行截图

状态: Accepted

源代码

```
#赵语涵2300012254
result = []
for i in input().split():
    result.append(i)
print(' '.join(result[::-1]))
```

基本信息

#: 44529295
题目: 27706
提交人: 23n2300012254
内存: 3608kB
时间: 26ms
语言: Python3
提交时间: 2024-04-04 19:00:36

27951: 机器翻译

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

思路：对于队列，用deque进行操作。对每个数字判断是否存在已有存储中，若无则计数+1。同时记录队列大小，当超过规定大小时注意放入后要弹出第一个，且size不再+1

代码

```
1  #赵语涵2300012254
2  from collections import deque
3  store = deque()
4  m,n = map(int,input().split())
5  size,count = 0,0
6  for i in map(int,input().split()):
7      if i not in store:
8          count += 1
9          store.append(i)
10         if size < m:
11             size += 1
12         else:
13             store.popleft()
14 print(count)
```

代码运行截图

状态: Accepted

源代码

```
#赵语涵2300012254
from collections import deque
store = deque()
m,n = map(int,input().split())
size,count = 0,0
for i in map(int,input().split()):
```

基本信息

#: 44529409
题目: 27951
提交人: 23n2300012254
内存: 3624kB
时间: 26ms
语言: Python3
提交时间: 2024-04-04 19:14:39

27932: Less or Equal

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

思路：题目比较简单，直接sort排序后看第k个数字是否是最后一个数字或者其后数字不等于它即可。但是要注意一些坑点：1) k可能=0，此时最小整数应该是当前最小数字-1；2) k=0的同时，最小数字可能为1，而要求输出数字x>0，则应该输出-1

代码

```
1  #赵语涵2300012254
2  n,k = map(int,input().split())
3  nums = list(map(int,input().split()))
4  nums.sort()
5  x = nums[k-1]
6  if k == 0:
7      if nums[0] != 1:
8          print(nums[0]-1)
9      else:
10         print(-1)
11 elif k == n:
12     print(x)
13 elif nums[k] != x:
14     print(x)
15 else:
16     print(-1)
```

代码运行截图

#44529526提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```
#赵语涵2300012254
n,k = map(int,input().split())
nums = list(map(int,input().split()))
nums.sort()
x = nums[k-1]
if k == 0:
    if nums[0] != 1:
```

基本信息

#: 44529526
题目: 27932
提交人: 23n2300012254
内存: 9904kB
时间: 45ms
语言: Python3
提交时间: 2024-04-04 19:31:42

27948: FBI树

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

思路：建立树类型以后，通过递归，分别输出左右子树和自身的类型（左右子树下如果还有子树就可以递归）。

代码

```
1 #赵语涵2300012254
2 def type(x):
3     if '1' not in x:
4         return 'B'
5     if '0' not in x:
6         return 'I'
7     return 'F'
8
9 class Node():
10     def __init__(self,x):
11         self.value = x
12         self.type = type(x)
13         self.left = None
14         self.right = None
15
16 def tree(node):
17     l = len(node.value)
18     if l==1:
19         return node.type
20     node.left = Node(node.value[:l//2])
21     node.right = Node(node.value[l//2:])
22     return tree(node.left)+tree(node.right)+node.type
23
24 n = input()
25 print(tree(Node(input())))
```

代码运行截图

#44529661提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```
#赵语涵2300012254
def type(x):
    if '1' not in x:
        return 'B'
    if '0' not in x:
        return 'I'
```

基本信息

#: 44529661
题目: 27948
提交人: 23n2300012254
内存: 3960kB
时间: 25ms
语言: Python3
提交时间: 2024-04-04 19:48:05

27925: 小组队列

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

思路：对于不同小组分别建立队列，只需要存储各小组第一个人进入队列的顺序（seq）以及各小组进入队列的当前人数即可。只有在前面的小组全部弹出以后第二个小组才会开始弹出

代码

```
1 #赵语涵2300012254
2 from collections import deque
3 des = dict()
4 for i in range((n:=int(input()))):
5     des[i] = deque()
6 contain,seq,count = dict(),deque(),dict(zip(range(n),[0]*n))
7 for i in range(n):
8     contain[i] = list(map(int,input().split()))
9
10 while True:
11     if (x:=input()) == 'STOP':
12         break
13     if x == 'DEQUEUE':
14         if count[(i:=seq[0])]>0:
15             print(des[i].popleft())
16             count[i] -= 1
17         else:
18             seq.popleft()
19             print(des[(i:=seq[0])].popleft())
20             count[i] -= 1
21     else:
22         ope,num = x.split()
23         num = int(num)
24         for i in range(n):
25             if num in contain[i]:
26                 des[i].append(num)
27                 count[i]+=1
28                 break
29         if i not in seq:
30             seq.append(i)
```

代码运行截图

#44529962提交状态

[查看](#) [提交](#) [统计](#)

状态: **Accepted**

源代码

```
#赵语涵2300012254
from collections import deque
des = dict()
for i in range((n:=int(input()))):
    des[i] = deque()
contain,seq,count = dict(),deque(),dict(zip(range(n),[0]*n))
```

基本信息

#: 44529962
题目: 27925
提交人: 23n2300012254
内存: 4288kB
时间: 1130ms
语言: Python3
提交时间: 2024-04-04 20:24:34

27928: 遍历树

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

思路：考试的时候试图使用树并记录树的父节点，通过递归处理。问题在于父节点在当前最小输出以后不再对父节点进行比较，自己的代码大概也就是在这里WA的。

参考了同学的代码，首先关于上面的问题，在递归时将父节点子节点类似地进行处理（代码中是父节点索引的是父节点和所有子节点的列表）；另外发现用树的方法也可以用字典实现，可能会使代码更简洁

代码

```
1  #赵语涵2300012254
2  tree={}
3  n=int(input())
4
5  def output(n):
6      for i in tree[n]:
7          if i == n:
8              print(i)
9          else:
10             output(i)
11
12  child=set()
13  record=set()
14  for i in range(n):
15      num=list(map(int,input().split()))
16      tree[num[0]]=sorted(num)
17      for ind,x in zip(range(len(num)),num):
18          if ind != 0:
19              child.add(x)
20              record.add(x)
21  root=record.difference(child).pop()
22  output(root)
```

代码运行截图

#44587783提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```
#赵语涵2300012254
tree={}
n=int(input())

def output(n):
    for i in tree[n]:
        if i == n:
```

基本信息

#: 44587783
题目: 27928
提交人: 23n2300012254
内存: 3768kB
时间: 36ms
语言: Python3
提交时间: 2024-04-09 21:24:46

2. 学习总结和收获

考试过程中前面4道题做起来比较顺利，用了将近1个小时做完（但是中途还是有一些小的卡bug比如 Less or Equal中的坑点之类的）；第5道题最开始想得过于复杂了，当理解到弹出顺序之和小组第一个人进入的顺序相关时就比较快的AC了。最后一道题感觉再有点时间就能debug了（最后是样例通过但是WA的情况）

总体来讲前面的题做起来比较轻松但是在细节上可能出现一些问题拖慢了时间，然后要多注意题干信息确认有没有坑点，后面的题应该要先理清解决思路再开始写的话可能会高效得多（自己有时候习惯边敲边想...）。