

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by ==王一粟 经济学院==

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：直接逆向输出即可；当然也可以通过栈的方式

耗时：3min

代码

```
#2200015507 王一粟
s = input().split()
s1 = s[::-1]
```

```
print(" ".join(s1))
```

代码运行截图 == (至少包含有"Accepted") ==

#44561844提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
s = input().split()
s1 = s[::-1]
print(" ".join(s1))
```

基本信息

#: 44561844

题目: 27706

提交人: 2200015507-王一粟

内存: 3592kB

时间: 26ms

语言: Python3

提交时间: 2024-04-07 15:11:53

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

27951: 机器翻译

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

思路：采用队列数据结构。当队列中没有时，加入该单词；注意这里队列有最大容量

耗时：6min

代码

```
#2200015507 王一粟
from collections import deque
m,n = [int(i) for i in input().split()]
passage = input().split()
d = deque()
cnt = 0
result = 0
for element in passage:
    if element not in d:
        result += 1
        if cnt < m:
            d.append(element)
            cnt +=1
        else:
            d.popleft()
            d.append(element)
print(result)
```

代码运行截图 == (至少包含有"Accepted") ==

#44561907提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from collections import deque
m,n = [int(i) for i in input().split()]
passage = input().split()
d = deque()
cnt = 0
result = 0
for element in passage:
    if element not in d:
        result += 1
        if cnt < m:
            d.append(element)
            cnt +=1
        else:
            d.popleft()
            d.append(element)
print(result)
```

基本信息

#: 44561907
题目: 27951
提交人: 2200015507-王一粟
内存: 3628kB
时间: 30ms
语言: Python3
提交时间: 2024-04-07 15:17:12

27932: Less or Equal

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

思路：采用堆结构输出前k个较小的数。如果第k+1个数（在有的情况下）与第k个数不同，则输出第k个数，反之不存在。第一次代码RE，原因在于没有考虑k=0的特殊情况

耗时：13min

代码

```
#2200015507 王一粟
import heapq
n,k = [int(i) for i in input().split()]
mylist = [int(i) for i in input().split()]
heapq.heapify(mylist)
if k == 0:
    t = heapq.heappop(mylist)
    if t == 1:
        print(-1)
    else:
        print(1)
else:
    for element in range(k):
        t = heapq.heappop(mylist)
    if k == n:
        print(t)
    else:
        m = heapq.heappop(mylist)
        if m == t:
            print(-1)
        else:
```

```
print(t)
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: **Accepted**

源代码

```
import heapq
n,k = [int(i) for i in input().split()]
mylist = [int(i) for i in input().split()]
heapq.heapify(mylist)
if k == 0:
    t = heapq.heappop(mylist)
    if t == 1:
        print(-1)
    else:
        print(1)
else:
    for element in range(k):
        t = heapq.heappop(mylist)
        if k == n:
            print(t)
        else:
            m = heapq.heappop(mylist)
            if m == t:
                print(-1)
            else:
                print(t)
```

基本信息

#: 44562036
题目: 27932
提交人: 2200015507-王一粟
内存: 10364kB
时间: 55ms
语言: Python3
提交时间: 2024-04-07 15:29:45

27948: FBI树

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

思路: 先解析二叉树: 序列长度为1直接判断B/I, 否则分别解析左、右子树并返回左右子树根结点的value, 再通过判断确定该结点的value。然后后序输出即可

耗时: 25min

代码

#2200015507 王一粟

```
class Node:
    def __init__(self):
        self.val = None
        self.left = None
        self.right = None
    def get(self):
        return self.val

def parse(s,n):
    if n==0:
        if s == "0":
            current_node = Node()
            current_node.val = "B"
            current_node.left,current_node.right = None,None
```

```

        return current_node
    else:
        current_node = Node()
        current_node.val = "I"
        current_node.left, current_node.right = None, None
        return current_node
    else:
        current_node = Node()
        current_node.left = parse(s[:2**(n-1)],n-1)
        current_node.right = parse(s[2**(n-1):],n-1)

        if current_node.left.val == "B" and current_node.right.val == "B":
            current_node.val = "B"
        elif current_node.left.val == "I" and current_node.right.val == "I":
            current_node.val = "I"
        else:
            current_node.val = "F"
        return current_node
def post(root):
    if root is None:
        return ""
    else:
        return post(root.left)+post(root.right)+root.get()
n = int(input())
s = input()
root = parse(s,n)
print(post(root))

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#44562481提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

class Node:
    def __init__(self):
        self.val = None
        self.left = None
        self.right = None
    def get(self):
        return self.val

def parse(s,n):
    if n==0:
        if s == "0":
            current_node = Node()
            current node.val = "B"

```

基本信息

#: 44562481
 题目: 27948
 提交人: 2200015507-王一粟
 内存: 4152kB
 时间: 26ms
 语言: Python3
 提交时间: 2024-04-07 15:56:44

27925: 小组队列

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

思路：我的思路是通过一个字典确定编号到小组的映射；一个queue装队列中小组编号（小组中人必然在一起），另新开一个队列索引每个小组中在队列里的成员。我做这道题是老师4.7下午改了之后的。感觉题目还是有点表述的不太清晰，因为题目并没有说可能有非小组成员加入队列的情况，也没有指明这种情况下应该怎么处

理（加入队列还是忽略）。因此WA&RE了四次，最后是通过try-except进行判断，考虑了非小组成员加入队列、以及在队列中没有人时dequeue的情况。

耗时：35min

代码

```
#2200015507 王一粟
from collections import deque
t = int(input())
mydict = {}
for i in range(1,t+1):
    s = input().split()
    for element in s:
        mydict[element] = i
d = deque()
queue = [deque() for i in range(t+1)]
while True:
    s = input()
    if s == "STOP":
        break
    if s == "DEQUEUE":
        try:
            try:
                idx = d[0]
            except:
                continue
            num = queue[idx].popleft()
            print(num)
            if queue[idx] == deque():
                d.popleft()
        except:
            print(idx)
            d.popleft()
    else:
        t,num = s.split()
        try:
            idx = mydict[num]
            if queue[idx] == deque():
                queue[idx].append(num)
                d.append(idx)
            else:
                queue[idx].append(num)
        except:
            d.append(num)
```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

#44563096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())
mydict = {}
for i in range(1,t+1):
    s = input().split()
    for element in s:
        mydict[element] = i
d = deque()
queue = [deque() for i in range(t+1)]
while True:
    s = input()
    if s == "STOP":
        break
    if s == "DEQUEUE":
        try:
            try:
```

基本信息

#: 44563096
题目: 27925
提交人: 2200015507-王一粟
内存: 5260kB
时间: 100ms
语言: Python3
提交时间: 2024-04-07 16:32:39

27928: 遍历树

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

思路：我的想法是首先把信息都输入进程序，通过一个列表先把每行开头的结点元素设置出来。然后再分别对每行遍历，进行树的解析，同时对于子结点进行标记。然后，通过一次循环找到根结点。接下来采用递归的方式进行序列输出。具体地，通过sort排列比较root与children的值，然后依次进行递归输出。代码在效率上稍显笨拙，感觉题目没有特别卡时间。

耗时：35min

代码

#2200015507 王一粟

```
class Node:
    def __init__(self,val):
        self.val = val
        self.child = []
    def chil(self):
        return self.child
    def va(self):
        return self.val
n = int(input())
data_list = []
node_dict = {}
for i in range(n):
    mylist = [int(i) for i in input().split()]
    node_dict[mylist[0]] = [Node(mylist[0]),1]
    data_list.append(mylist)
for p in data_list:
    for i in p[1:]:
        node_dict[p[0]][0].child.append(node_dict[i][0])
        node_dict[i][1] = 0
```

```

for element,k in node_dict.items():
    if k[1] == 1:
        root = k[0]
        break
def parse(root):
    result = []
    if root.chil() == []:
        return [root.va()]

    operate = sorted(root.chil()+[root],key = lambda x:x.va())
    for element in operate:
        if element == root:
            result.append(root.va())
        else:
            result += parse(element)
    return result
t = parse(root)
for i in t:
    print(int(i))

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: **Accepted**

源代码

```

class Node:
    def __init__(self, val):
        self.val = val
        self.child = []
    def chil(self):
        return self.child
    def va(self):
        return self.val

n = int(input())
data_list = []
node_dict = {}
for i in range(n):
    mylist = [int(i) for i in input().split()]
    node_dict[mylist[0]] = [Node(mylist[0]), 1]
    data_list.append(mylist)
for p in data_list:

```

基本信息

#: 44563686
 题目: 27928
 提交人: 2200015507-王一粟
 内存: 3868kB
 时间: 25ms
 语言: Python3
 提交时间: 2024-04-07 17:10:51

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。==

本周新知识点整体不多, 主要是偏概念的散列这部分内容, 学这块花费的时间不多;

课堂笔试, 感觉自己还是有挺多对概念理解不完备的地方的, 最后期末前肯定还要再做系统性梳理;

自己这周有点忙, 今天(周日)下午去机房掐时间做了下月考的六道题。规定时间内是AC5, 其中小组队列那个被enqueue非小组成员的搞了一下。最后一道题AC比规定时间晚了5分钟左右, 可能对于难题的做题速度还有待

提高，再有就是第一次编码总是会漏一些边角的情况，以及机房和自己笔记本的键盘还是挺有差距、打字速度有点慢xs

补课的话，老师dp的讲义看完了，感觉dp有丢丢难，不过常规的思路大致懂了

接下来期中季每日选做可能要暂时停一下，等4月中旬考完之后再补，第九周三天五门压力有点大