

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled 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: stack, 感谢送分

代码

```
1 raw=list(map(str,input().split()))
2 stack=[]
3 while raw:
4     stack.append(raw.pop())
5 print(' '.join(stack))
```

代码运行截图 (至少包含有"Accepted")

#44516166提交状态

状态: Accepted

源代码

```
raw=list(map(str,input().split()))
stack=[]
while raw:
    stack.append(raw.pop())
print(' '.join(stack))
```

©2002-2022 POJ 京ICP备20010980号-1

27951: 机器翻译

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

思路:

代码

```
1 from collections import deque
2 dic,dic_set,limit,cnt=deque(),set(),0,0
3 M,N=map(int,input().split())
4 passage=list(map(int,input().split()))
5
6 for i in passage:
7     if limit<=M-1:
8         if i in dic_set:continue
9         else:
10             cnt+=1
11             limit+=1
12             dic.append(i)
13             dic_set.add(i)
14     else:
15         if i in dic_set:continue
16         else:
17             tmp=dic.popleft()
18             dic_set.discard(tmp)
19             dic.append(i)
20             dic_set.add(i)
21             cnt+=1
22 print(cnt)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

基

源代码

```
from collections import deque
dic,dic_set,limit,cnt=deque(),set(),0,0
M,N=map(int,input().split())
passage=list(map(int,input().split()))

for i in passage:
    if limit<=M-1:
        if i in dic_set:continue
        else:
            cnt+=1
            limit+=1
            dic.append(i)
            dic_set.add(i)
    else:
        if i in dic_set:continue
        else:
            tmp=dic.popleft()
            dic_set.discard(tmp)
            dic.append(i)
            dic_set.add(i)
            cnt+=1

print(cnt)
```

27932: Less or Equal

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

思路:

代码

```
1 n,k=map(int,input().split())
2 raw=list(map(int,input().split()))
3 raw.sort()
4 x=None
5 if k==0:
6     if raw[0]==1:x=-1
7     else:x=raw[0]-1
8
9 elif k>=1 and k<n:
10     if raw[k]>raw[k-1]:
11         x=raw[k-1]
12     else:
13         x=-1
14 else:x=raw[-1]
15
16 if x===-1:
17     print(x)
18 elif x>=1 and x<=10**9:
19     print(x)
20 else:
21     print(-1)
```

#44517870提交状态

状态: Accepted

源代码

```
n,k=map(int,input().split())
raw=list(map(int,input().split()))
raw.sort()
x=None
if k==0:
    if raw[0]==1:x=-1
    else:x=raw[0]-1

elif k>=1 and k<n:
    if raw[k]>raw[k-1]:
        x=raw[k-1]
    else:
        x=-1
else:x=raw[-1]

if x===-1:
    print(x)
elif x>=1 and x<=10**9:
    print(x)
else:
```

27948: FBI树

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

思路:

和归并排序有相似之处。

由于这是一颗完美的二叉树, 所以可以考虑从叶子结点向上建树。

代码

```
1  """我们可以把由 0 和 1 组成的字符串分为三类: 全 0 串称为 B 串, 全 1 串称为 I 串
2  既含 0 又含 1 的串则称为 F 串。
3  FBI 树是一种二叉树, 它的结点类型也包括 F 结点, B 结点和 I 结点三种。"""
4  class TreeNode:
5      def __init__(self,val):
6          self.val=val
7          self.left=None
8          self.right=None
9
10 N=int(input())
11 raw=list(input())
12 ltype=[]
13 for i in raw:
14     if i=='1':ltype.append(TreeNode('I'))
15     else:ltype.append(TreeNode('B'))
```

```

16
17 def parent_type(type1,type2):
18     if type1==type2=='I':
19         return 'I'
20     elif type1==type2=='B':
21         return "B"
22     else:return "F"
23
24 while len(ltype)>1:
25     tmp=[]
26     for i in range(0,len(ltype),2):
27         left=ltype[i]
28         right=ltype[i+1]
29         parent=TreeNode(parent_type(left.val,right.val))
30         parent.left=left
31         parent.right=right
32         tmp.append(parent)
33     ltype=tmp
34
35 def post(node):
36     output=[]
37     if node==None:return output
38     output.extend(post(node.left))
39     output.extend(post(node.right))
40     output.append(node.val)
41     return "".join(output)
42
43 print(post(ltype[0]))

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

基

源代码

```
"""我们可以把由 0 和 1 组成的字符串分为三类：全 0 串称为 B 串，全 1 串称为 I 串
既含 0 又含 1 的串则称为 F 串。
FBI 树是一种二叉树，它的结点类型也包括 F 结点，B 结点和 I 结点三种。"""
class TreeNode:
    def __init__(self, val):
        self.val=val
        self.left=None
        self.right=None

N=int(input())
raw=list(input())
ltype=[]
for i in raw:
    if i=='1':ltype.append(TreeNode('I'))
    else:ltype.append(TreeNode('B'))

def parent_type(type1,type2):
    if type1==type2=='I':
        return 'I'
    elif type1==type2=='B':
        return "B"
    else:return "F"

while len(ltype)>1:
    tmp=[]
    for i in range(0,len(ltype),2):
        left=ltype[i]
        right=ltype[i+1]
        parent=TreeNode(parent_type(left.val,right.val))
```

基

27925: 小组队列

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

思路:

维护一个队列序s_seq, 由队列序生成一个到队列的映射。

代码

```
1 from collections import defaultdict,deque
2 class node_deque:
3     def __init__(self,idx):
4         self.idx=idx
5         self.queue=deque()
6     # 多个映射, 由队列索引映射到节点, 以及由队员映射到索引
7     idx_to_node,member_to_idx,ans,q_seq={},defaultdict(int),[],deque()
8     t=int(input())
9
10    for i in range(t):
11        idx_to_node[i+1]=node_deque(i+1)
12        for member in map(int,input().split()):
13            member_to_idx[member]=(i+1)
14
15    while (command:=input())!='STOP':
16        if command[0]=='D':
```

```

17         ans.append(idx_to_node[q_seq[0]].queue.popleft())
18         if not idx_to_node[q_seq[0]].queue: q_seq.popleft()
19
20     elif command[0]=='E':
21         a,b=map(str,command.split())
22         idx=member_to_idx[int(b)]
23         if idx_to_node[idx].queue:
24             idx_to_node[idx].queue.append(b)
25         else:
26             q_seq.append(idx)
27             idx_to_node[idx].queue.append(b)
28
29 for i in ans:
30     print(i)

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

去

源代码

```

from collections import defaultdict, deque
class node_deque:
    def __init__(self, idx):
        self.idx=idx
        self.queue=deque()
# 多个映射, 由队列索引映射到节点, 以及由队员映射到索引
idx_to_node, member_to_idx, ans, q_seq={}, defaultdict(int), [], deque()
t=int(input())

for i in range(t):
    idx_to_node[i+1]=node_deque(i+1)
    for member in map(int, input().split()):
        member_to_idx[member]=(i+1)

while (command:=input())!='STOP':
    if command[0]=='D':
        ans.append(idx_to_node[q_seq[0]].queue.popleft())
        if not idx_to_node[q_seq[0]].queue: q_seq.popleft()

    elif command[0]=='E':
        a,b=map(str,command.split())
        idx=member_to_idx[int(b)]
        if idx_to_node[idx].queue:
            idx_to_node[idx].queue.append(b)
        else:
            q_seq.append(idx)
            idx_to_node[idx].queue.append(b)

for i in ans:
    print(i)

```

27928: 遍历树

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

思路:

代码

```
1 class TreeNode:
2     def __init__(self, val):
3         self.val = val
4         self.children = []
5         self.parent = None
6
7 def get_val(node: TreeNode):
8     tmp = []
9     output = []
10    if node == None: return output
11    if not node.children: return [node.val]
12
13    tmp.append(node)
14    tmp.extend(node.children)
15    tmp.sort(key=lambda x: x.val)
16
17    for son in tmp:
18        if son == node:
19            output.append(son.val)
20        else:
21            output.extend(get_val(son))
22    return output
23
24 n = int(input())
25 tree_dict = {}
26 for i in range(n):
27     a = input()
28     if len(a) == 1:
29         try:
30             tmp = tree_dict[int(a)]
31         except:
32             tmp = TreeNode(int(a))
33             tree_dict[int(a)] = tmp
34     else:
35         lst = list(map(int, a.split()))
36         val = lst[0]
37         try: tmp = tree_dict[val]
38         except: tree_dict[val] = TreeNode(val)
39
40         for i in lst[1:]:
41             try: tree_dict[i].parent = tree_dict[val]
42             except:
43                 tree_dict[i] = TreeNode(i)
44                 tree_dict[i].parent = tree_dict[val]
45
46         tree_dict[val].children.extend(tree_dict[i] for i in lst[1:])
47
48
```



```
49 root=None
50 for node in tree_dict.values():
51     if node.parent==None:
52         root=node
53         break
54 for i in get_val(root):
55     print(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 TreeNode:
    def __init__(self, val):
        self.val=val
        self.children=[]
        self.parent=None

def get_val(node:TreeNode):
    tmp=[]
    output=[]
    if node==None: return output
    if not node.children: return [node.val]

    tmp.append(node)
    tmp.extend(node.children)
    tmp.sort(key=lambda x:x.val)

    for son in tmp:
        if son==node:
            output.append(son.val)
        else:
            output.extend(get_val(son))
    return output

n=int(input())
tree_dict={}

```

2. 学习总结和收获

没睡醒, 痛失半小时coding时间, 差一道题AC6, 这是离AC6最近的一次!

题目难度不大, 模板性很强, 平时写习惯了很快就能AC。

学会了用pycharm debug以后上机快乐翻倍! 半分钟内发现bug, 大量节省时间。

笔试题目不熟悉, 说明只学会了使用接口, 内部实现理解的还不透彻, 抓紧补上。