

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by 陈奕好 工学院

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 Sonoma 14.4 (23E214)

Python编程环境：PyCharm 2023.3.1 (Professional Edition)

1. 题目

27706: 逐词倒放

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

思路：简单题

代码

```
1 arr = list(map(str,input().split()))
2 print(*arr[::-1])
3
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
arr = list(map(str, input().split()))
print(*arr[::-1])
```

基本信息

#: 44523595
题目: 27706
提交人: 23n2300011030(陈奕好)
内存: 3576kB
时间: 25ms
语言: Python3
提交时间: 2024-04-03 23:45:06

27951: 机器翻译

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

思路: 滑动窗口

代码

```
1  from collections import deque
2  M, N = map(int, input().split())
3  arr = list(map(int, input().split()))
4  window = deque()
5  ans = 0
6  for i in arr:
7      if i in window:
8          continue
9      else:
10         ans += 1
11         if len(window) == M:
12             window.popleft()
13         window.append(i)
14 print(ans)
15
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
M, N = map(int, input().split())
arr = list(map(int, input().split()))
window = deque()
ans = 0
for i in arr:
    if i in window:
        continue
    else:
        ans += 1
        if len(window) == M:
            window.popleft()
        window.append(i)
print(ans)
```

基本信息

#: 44523658

题目: 27951

提交人: 23n2300011030(陈奕好)

内存: 3632kB

时间: 23ms

语言: Python3

提交时间: 2024-04-03 23:56:49

27932: Less or Equal

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

思路: 感觉很trick, 这个代码需要微调

代码

```
1  n, k = map(int, input().split())
2  arr = list(map(int, input().split()))
3  arr.sort()
4  if k == 0:
5      if arr[0] > 1:
6          print(1)
7      else:
8          print(-1)
9  elif k == n or arr[k-1] < arr[k]:
10     print(arr[k-1])
11 else:
12     print(-1)
13
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

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

基本信息

#: 44523723
题目: 27932
提交人: 23n2300011030(陈奕好)
内存: 9804kB
时间: 44ms
语言: Python3
提交时间: 2024-04-04 00:08:19

27948: FBI树

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

思路: FBI! 这道题我问了几乎所有的ai, 但几乎都过不了, 甚至是一个很小的、可查的bug。判断FBI的板块死活放不对位置。

代码

```
1 class TreeNode:
2     def __init__(self, value):
3         self.val = value
4         self.left = None
5         self.right = None
6
7 def build_tree(s):
8     if len(s) == 1:
9         return TreeNode(s)
10    top = TreeNode(s)
11    top.left = build_tree(s[:len(s)//2])
12    top.right = build_tree(s[len(s)//2:])
13    return top
14
15
16 def postorder_traversal(root):
17     if root is None:
18         return ''
19     left = postorder_traversal(root.left)
20     right = postorder_traversal(root.right)
21     if set(root.val) == {'0'}:
22         value = 'B'
23     elif set(root.val) == {'1'}:
```

```

24         value = 'I'
25     else:
26         value = 'F'
27     return left + right + value
28
29
30 N = int(input())
31 print(postorder_traversal(build_tree(input())))
32

```

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, value):
        self.val = value
        self.left = None
        self.right = None

def build_tree(s):
    if len(s) == 1:
        return TreeNode(s)
    top = TreeNode(s)
    top.left = build_tree(s[:len(s)//2])
    top.right = build_tree(s[len(s)//2:])
    return top

def postorder_traversal(root):
    if root is None:
        return ''
    left = postorder_traversal(root.left)
    right = postorder_traversal(root.right)
    if set(root.val) == {'0'}:
        value = 'B'
    elif set(root.val) == {'1'}:
        value = 'I'
    else:
        value = 'F'
    return left + right + value

N = int(input())
print(postorder_traversal(build_tree(input())))

```

基本信息

#: 44520452
 题目: 27948
 提交人: 23n2300011030(陈奕好)
 内存: 3984kB
 时间: 27ms
 语言: Python3
 提交时间: 2024-04-03 18:28:11

27925: 小组队列

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

思路: 字典里套用deque()的想法第一次实现

代码

```

1  from collections import deque
2
3  t = int(input())
4  teams = {i: deque(map(int, input().split())) for i in range(t)}
5  queue = deque()
6  group_queue = {i: deque() for i in range(t)}
7
8  while True:
9      command = input().split()
10     if command[0] == 'STOP':
11         break
12     elif command[0] == 'ENQUEUE':
13         person = int(command[1])
14         for i in range(t):
15             if person in teams[i]:
16                 group_queue[i].append(person)
17                 if i not in queue:
18                     queue.append(i)
19                 break
20     elif command[0] == 'DEQUEUE':
21         group = queue[0]
22         print(group_queue[group].popleft())
23         if not group_queue[group]:
24             queue.popleft()
25
26

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

基本信息

```
from collections import deque

t = int(input())
teams = {i: deque(map(int, input().split())) for i in range(t)}
queue = deque()
group_queue = {i: deque() for i in range(t)}

while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        person = int(command[1])
        for i in range(t):
            if person in teams[i]:
                group_queue[i].append(person)
                if i not in queue:
                    queue.append(i)
                break
    elif command[0] == 'DEQUEUE':
        group = queue[0]
        print(group_queue[group].popleft())
        if not group_queue[group]:
            queue.popleft()
```

题目: 27925
是交人: 23n2300011030(陈奕好)
内存: 4200kB
时间: 1263ms
语言: Python3
交时间: 2024-04-04 00:17:00

```

1 class Node:
2     def __init__(self, value):
3         self.value = value
4         self.children = []
5
6     def add_node(nodes, parent, child):
7         if parent not in nodes:
8             nodes[parent] = Node(parent)
9         if child not in nodes:
10            nodes[child] = Node(child)
11            nodes[parent].children.append(nodes[child])
12
13    def traverse(node):
14        values = [node.value] + [child.value for child in node.children]

```

```
15     values.sort()
16     for value in values:
17         if value == node.value:
18             print(value)
19         else:
20             traverse(nodes[value])
21
22 # Parse the input
23 n = int(input())
24 nodes = {}
25 root = None
26 leaves = set()
27 for _ in range(n):
28     line = list(map(int, input().split()))
29     leaves |= set(line[1:])
30     parent = line[0]
31     if root is None:
32         root = parent
33     for child in line[1:]:
34         add_node(nodes, parent, child)
35
36 for i in nodes.values():
37     if i.value not in leaves:
38         root = i.value
39         break
40
41 # Traverse the tree
42 traverse(nodes[root])
43
```

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, value):
        self.value = value
        self.children = []

def add_node(nodes, parent, child):
    if parent not in nodes:
        nodes[parent] = Node(parent)
    if child not in nodes:
        nodes[child] = Node(child)
    nodes[parent].children.append(nodes[child])

def traverse(node):
    values = [node.value] + [child.value for child in node.children]
    values.sort()
    for value in values:
        if value == node.value:
            print(value)
        else:
            traverse(nodes[value])

# Parse the input
n = int(input())
nodes = {}
root = None
leaves = set()
for _ in range(n):
    line = list(map(int, input().split()))
```

基本信息

#: 44523569
题目: 27928
提交人: 23n2300011030(陈奕好)
内存: 3768kB
时间: 28ms
语言: Python3
提交时间: 2024-04-03 23:42:20

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。

没时间当堂考，字典重要性突然增强，之前一直被value与树的联系犯难，现在出现了这个东西。

$$list(dict[i]) \tag{1}$$

$$dict[i] = list \tag{2}$$

这是个可删除dict[i],方便对list进行管理，且 dict[i] 也可以是动态的。