

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by 夏天, 生命科学学院

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: Windows 10 家庭版

Python编程环境: Spyder (Python 3.11)

1. 题目

27706: 逐词倒放

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

思路: 很简单, 不解释

代码

```
1 sentence=input().split()
2 print(' '.join(sentence[::-1]))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
sentence=input().split()
print(' '.join(sentence[::-1]))
```

基本信息

#: 44515898
题目: E27706
提交人: 23n2300012289
内存: 3564kB
时间: 23ms
语言: Python3
提交时间: 2024-04-03 15:12:10

27951: 机器翻译

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

思路：用双端队列模拟由于内存限制单词被覆盖

代码

```
from collections import deque
M,N=map(int,input().split())
words=input().split()
queue=deque([])
count=0
for word in words:
    if word not in queue:
        if len(queue)==M:
            queue.popleft()
        queue.append(word)
        count+=1
print(count)
```

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())
words=input().split()
queue=deque([])
count=0
for word in words:
    if word not in queue:
        if len(queue)==M:
            queue.popleft()
        queue.append(word)
        count+=1
print(count)
```

基本信息
#: 44516073
题目: E27951
提交人: 23n2300012289
内存: 3640kB
时间: 24ms
语言: Python3
提交时间: 2024-04-03 15:18:52

27932: Less or Equal

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

思路：注意讨论k=0和k=n的特殊情况

代码

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

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

基本信息
#: 44518866
题目: M27932
提交人: 23n2300012289
内存: 9880kB
时间: 44ms
语言: Python3
提交时间: 2024-04-03 16:50:07

27948: FBI树

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

思路：根据题目要求套模板即可

代码

```
class FBITree:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None
def build_tree(string):
    if string == '1':
        return FBITree('I')
    if string == '0':
        return FBITree('B')
    if '1' in string and '0' in string:
        root = FBITree('F')
    elif '0' not in string:
        root = FBITree('I')
    else:
        root = FBITree('B')
    string1 = string[:len(string)//2]
    string2 = string[len(string)//2:]
    root.left = build_tree(string1)
    root.right = build_tree(string2)
    return root
def postfix(root):
    if not root:
        return ''
    result = ''
    result += postfix(root.left)
    result += postfix(root.right)
    result += root.val
    return result
N = int(input())
string = input()
root = build_tree(string)
result = postfix(root)
print(result)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
class FBITree:
    def __init__(self, val):
        self.val = val
        self.left = None
        self.right = None
def build_tree(string):
    if string == '1':
        return FBITree('I')
    if string == '0':
        return FBITree('B')
    if '1' in string and '0' in string:
        root = FBITree('F')
    elif '0' not in string:
        root = FBITree('I')
    else:
        root = FBITree('B')
    string1 = string[:len(string)//2]
    string2 = string[len(string)//2:]
    root.left = build_tree(string1)
    root.right = build_tree(string2)
    return root
def postfix(root):
    if not root:
        return ''
    result = ''
    result += postfix(root.left)
    result += postfix(root.right)
    result += root.val
    return result
N = int(input())
string = input()
root = build_tree(string)
result = postfix(root)
print(result)
```

基本信息

#: 44517330
题目: M27948
提交人: 23n2300012289
内存: 3856KB
时间: 26ms
语言: Python3
提交时间: 2024-04-03 16:00:38

27925: 小组队列

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

思路: 哪个小组里的第一个人入队, 最后出队时该小组一定先出完, 故只需要记录小组进入队列的先后顺序和每个小组中成员入队的先后顺序

代码

```
from collections import deque
t = int(input())
groups_dict = {}
members_dict = {}
for _ in range(t):
    members = list(map(int, input().split()))
    groups_dict[_+1] = deque()
    for member in members:
        members_dict[member] = _+1
queue = deque()
queue_set = set()
while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        person = int(command[1])
        group = members_dict[person]
        groups_dict[group].append(person)
        if group not in queue_set:
            queue.append(group)
            queue_set.add(group)
    else:
        if queue:
            group = queue[0]
            person = groups_dict[group].popleft()
            print(person)
            if not groups_dict[group]:
                queue.popleft()
                queue_set.remove(group)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
t = int(input())
groups_dict = {}
members_dict = {}
for _ in range(t):
    members = list(map(int, input().split()))
    groups_dict[_+1] = deque()
    for member in members:
        members_dict[member] = _+1
queue = deque()
queue_set = set()
while True:
    command = input().split()
    if command[0] == 'STOP':
        break
    elif command[0] == 'ENQUEUE':
        person = int(command[1])
        group = members_dict[person]
        groups_dict[group].append(person)
        if group not in queue_set:
            queue.append(group)
            queue_set.add(group)
    else:
        if queue:
            group = queue[0]
            person = groups_dict[group].popleft()
            print(person)
            if not groups_dict[group]:
                queue.popleft()
                queue_set.remove(group)
```

基本信息

#: 44522637
题目: 27925
提交人: 23n2300012289
内存: 4952KB
时间: 103ms
语言: Python3
提交时间: 2024-04-03 21:46:42

27928: 遍历树

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

思路: 用类写一个树, 然后dfs

代码

```
import heapq
class Treemode:
    def __init__(self, value):
        self.value = value
        self.child = []
        self.parent = None
    def __lt__(self, other):
        return int(self.value) < int(other.value)
```

```
def build_tree(nodes):
    nodes_dict={}
    for node_values in nodes:
        parent_value=node_values[0]
        if parent_value not in nodes_dict:
            nodes_dict[parent_value]=Treenode(parent_value)
        parent_node=nodes_dict[parent_value]
        child_values=node_values[1:]
        for child_value in child_values:
            if child_value not in nodes_dict:
                nodes_dict[child_value]=Treenode(child_value)
            child_node=nodes_dict[child_value]
            child_node.parent=parent_node
            parent_node.child.append(child_node)
    for node in nodes_dict.values():
        if not node.parent:
            root=node
            break
    return root
def traversal(root,result):
    if not root.child:
        result.append(root.value)
        return
    children=root.child
    temp=[root]+children
    temp.sort()
    for child in temp:
        if child==root:
            result.append(root.value)
        else:
            traversal(child,result)
n=int(input())
nodes=[input().split() for i in range(n)]
root=build_tree(nodes)
result=[]
traversal(root,result)
for _ in result:
    print(_)
```

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
class Treenode:
    def __init__(self,value):
        self.value=value
        self.child=[]
        self.parent=None
    def __lt__(self,other):
        return int(self.value)<int(other.value)
def build_tree(nodes):
    nodes_dict={}
    for node_values in nodes:
        parent_value=node_values[0]
        if parent_value not in nodes_dict:
            nodes_dict[parent_value]=Treenode(parent_value)
        parent_node=nodes_dict[parent_value]
        child_values=node_values[1:]
        for child_value in child_values:
            if child_value not in nodes_dict:
                nodes_dict[child_value]=Treenode(child_value)
            child_node=nodes_dict[child_value]
            child_node.parent=parent_node
            parent_node.child.append(child_node)
    for node in nodes_dict.values():
        if not node.parent:
            root=node
            break
    return root
def traversal(root,result):
    if not root.child:
        result.append(root.value)
        return
```

基本信息

#: 44522211
题目: 27928
提交人: 23h2300012289
内存: 3788kB
时间: 26ms
语言: Python3
提交时间: 2024-04-03 21:12:17

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。

本次月考AC4, E1, E2, M2很快就写出来了; M1其实也是, 但一开始没有考虑到边界情况导致WA, 临交卷的时候灵光乍现补上了。前4道题做完之后一直在磕T2 (T1看的时候没有清晰的思路, 遂跳过, 结果最后也没来得及再看一遍) 建完树之后不知道为什么根节点的孩子节点列表一直是空的, 考试结束后让gpt帮忙修改才知道正确的做法是用字典存节点, 而不是root=Treenode (根节点的值), 这样相当于创建了新的类, 原来的孩子节点列表就没有了; 其次, 遍历时应该用dfs, 然而我没意识到这件事, 看到大小关系想用堆写, 结果得到的输出不对。一句话总结就是要注意细节并且合理分配考试时间