

Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Compiled by 赵策 数学科学学院

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

操作系统: Windows 11

Python编程环境: Visual Studio Code 1.86.2

1. 题目

27706: 逐词倒放

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

思路:

代码

```
#
words=input().split()
words.reverse()
print(' '.join(words))
```

代码运行截图 (至少包含有"Accepted")

27706: 逐词倒放

Accepted

27951: 机器翻译

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

思路:

代码

```
#
m,n=map(int,input().split())
words=input().split()
mem=[]
cnt=0
for i in words:
    if i not in mem:
        mem.append(i)
        cnt+=1
    if len(mem)>m:
        mem.pop(0)
print(cnt)
```

代码运行截图 (至少包含有"Accepted")

27951: 机器翻译

Accepted

27932: Less or Equal

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

思路:

代码

```
#
n,k=map(int,input().split())
sequence=list(map(int,input().split()))
if k==0:
    if 1 in sequence:
        print(-1)
    else:
        print(1)
else:
    sequence.sort()
    sgn=0
    for i in sequence:
        if i==sequence[k-1]:
            sgn+=1
```

```
if sgn==1:
    print(sequence[k-1])
else:
    print(-1)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

27932: Less or Equal

Accepted

27948: FBI树

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

思路:

代码

```
#
def fbi(s):
    if set(list(s))=={'0'}:
        return 'B'
    elif set(list(s))=={'1'}:
        return 'I'
    else:
        return 'F'

def build_tree(s):
    if len(s)==1:
        return fbi(s)
    else:
        middle=len(s)//2
        left_substring=s[:middle]
        right_substring=s[middle:]
        root_type=fbi(s)
        T1=build_tree(left_substring)
        T2=build_tree(right_substring)
        return T1+T2+root_type

N=int(input())
binary_string=input()
fbi_tree_postorder=build_tree(binary_string)
print(fbi_tree_postorder)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

27948: FBI树

Accepted

27925: 小组队列

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

思路:

代码

```
#
def simulate_queue(groups, command):
    if command[0]=='ENQUEUE':
        person=command[1]
        l=len(queue)
        if l in [0,1]:
            queue.append(person)
        else:
            i=0
            while i<l-1:
                if group_map[person]==group_map[queue[i]] and
group_map[person]!=group_map[queue[i+1]]:
                    queue.insert(i+1,person)
                    break
                else:
                    i+=1
            if i==l-1:
                queue.append(person)
    else:
        print(queue.pop(0))

queue=[]
group_map={}
groups=[input().split() for _ in range(int(input()))]
for group_index,group in enumerate(groups):
    for person in group:
        group_map[person]=group_index
while True:
    command=input().split()
    if command[0]=='STOP':
        break
    else:
        simulate_queue(groups,command)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

27925: 小组队列

Accepted

27928: 遍历树

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

思路:

代码

```
#
class TreeNode:
    def __init__(self,value):
        self.value=value
        self.children=[]

def sort_node(node):
    global res
    node.children.sort(key=lambda x:x.value)
    sgn=1
    for child in node.children:
        idx=res.index(node.value)
        if child.value<node.value:
            res.insert(idx,child.value)
        else:
            res.insert(idx+sgn,child.value)
            sgn+=1
    for child in node.children:
        sort_node(child)
    return res

nodes={}
for _ in range(int(input())):
    values=list(map(int,input().split()))
    node_val=values[0]
    children_vals=values[1:]
    if node_val not in nodes:
        nodes[node_val]=TreeNode(node_val)
    node=nodes[node_val]
    if children_vals:
        for child_val in children_vals:
            if child_val not in nodes:
                nodes[child_val]=TreeNode(child_val)
            child_node=nodes[child_val]
            node.children.append(child_node)

root=None
for node_val,node in nodes.items():
    is_root=True
    for _,other_node in nodes.items():
        if node_val in [child.value for child in other_node.children]:
            is_root=False
            break
    if is_root:
```

```
        root=node
        break

res=[root.value]
result=sort_node(root)
for val in result:
    print(val)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

27928: 遍历树

Accepted

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。

难度还行, 耗时不太长。除了最后一题最开始犯了低级错误, 自检发现了:

```
# 第一版
def sort_node(node):
    node.children.sort(key=lambda x:x.value)
    for child in node.children:
        idx=res.index(node.value)
        if child.value<node.value:
            res.insert(idx,child.value)
        else:
            res.insert(idx+1,child.value)
    for child in node.children:
        sort_node(child)
    return res

#大于的时候不能直接插到idx+1的位置。。

#修改后(不想大范围改动了。。)
def sort_node(node):
    node.children.sort(key=lambda x:x.value)
    sgn=1
    for child in node.children:
        idx=res.index(node.value)
        if child.value<node.value:
            res.insert(idx,child.value)
        else:
            res.insert(idx+sgn,child.value)
            sgn+=1
    for child in node.children:
        sort_node(child)
    return res
```