Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Complied by 城环 吴至超

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：Windows11

Python编程环境：Pycharm2023.2.3

1. 题目

27706: 逐词倒放

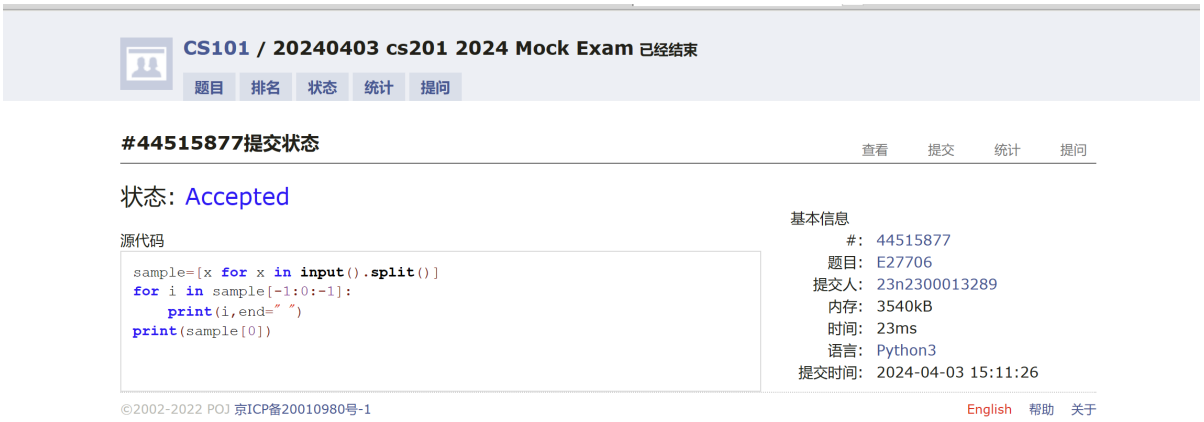
http://cs101.openjudge.cn/practice/27706/

思路：列表倒序

代码

```
#
sample=[x for x in input().split()]
for i in sample[-1:0:-1]:
    print(i,end=" ")
print(sample[0])
```

代码运行截图 ==（至少包含有"Accepted"）==



27951: 机器翻译

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

思路：用列表操作，每次看看需要的单词是否在内存中，如不在，更新内存即可。

代码

```
#
M,N=(int(x) for x in input().split())
lis=[int(x) for x in input().split()]
contain=[]
cnt=0
for i in lis:

    if i not in contain and len(contain)<M:
        contain.append(i)
        cnt+=1
    elif i not in contain and len(contain)==M:
        contain.pop(0)
        contain.append(i)
        cnt+=1
print(cnt)
```

代码运行截图 ==（至少包含有"Accepted"）==

27932: Less or Equal

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

思路：注意一下k=0的情况就好，此时应该取列表最小项-1，看看这个数是否满足范围。

代码

```
#
n,k=(int(x) for x in input().split())
lis=[int(x) for x in input().split()]
lis.sort()
if k!=0:
    a=lis[k-1]
    if a <= 0:
        print("1")
    elif a > 10 ** 9:
        print("-1")
    elif k < n and lis[k] == a:
        print("-1")
    elif k == n:
        print(a)
    elif k < n and lis[k] > a:
        print(a)
else:
    a=lis[k]-1
    if a>=1:
        print(a)
    else:
        print("-1")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

27948: FBI树

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

思路：就是正常的建树，有点像中序+前or后序建树

代码

```
#
N=int(input())

sample=input()
class tree:
    def __init__(self,char):
        self.char=char
        self.left=None
        self.right=None
def buildtree(root):#root是字符串
    length=len(root)
    if length>1:
        node=None
        if root =="":
            return None
        if "1" in root and "0" in root:
            node=tree("F")
        elif "0" not in root and "1" in root:
            node=tree("I")
```

```
        elif "1" not in root and "0" in root:
            node=tree("B")
        left=root[0:length//2]
        right=root[length//2::]
        node.left=buildtree(left)
        node.right=buildtree(right)
        return node
    else:
        if root=="1":
            node=tree("I")
        else:
            node=tree("B")
        return node

def postorder(node):
    stack=[]
    if node.left:
        stack.extend(postorder(node.left))
    if node.right:
        stack.extend(postorder(node.right))
    stack.append(node.char)
    return stack

print("".join(postorder(buildtree(sample))))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44517721提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
N=int(input())

sample=input()
class tree:
    def __init__(self,char):
        self.char=char
        self.left=None
        self.right=None
def buildtree(root):#root是字符串
    length=len(root)
    if length>1:
        node=None
        if root =="":
            return None
        if "1" in root and "0" in root:
            node=tree("F")
        elif "0" not in root and "1" in root:
            node=tree("I")
        elif "1" not in root and "0" in root:
            node=tree("B")
        left=root[0:length//2]
        right=root[length//2::]
        node.left=buildtree(left)
        node.right=buildtree(right)
        return node
    else:
        if root=="1":
            node=tree("I")
        else:
            node=tree("B")
        return node

def postorder(node):
```

基本信息

#: 44517721
题目: M27948
提交人: 23n2300013289
内存: 3908kB
时间: 25ms
语言: Python3
提交时间: 2024-04-03 16:15:41

27925: 小组队列

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

思路：用了字典（key为“1”，“2”...）来快速判别当前队列里有没有组员，当前队列里的每一个元素为[组员的编号，组别（1，2...）]，用str转换一下就好。

代码

```
#
t=int(input())
groups= {}
case=1
for i in range(t):
    a=[int(x) for x in input().split()]
    for m in a:
        groups[str(m)]=case
    case+=1
queue=[]
while True:
    try:

        order=[x for x in input().split()]
        if order[0]=="STOP":
            break
        if order[0]=="ENQUEUE":
            if not queue:
                queue.append([order[1],groups[order[1]]])#字符串，数字;人员编号，组别
            else:
                pan=groups[order[1]]#组号
                flag=0
                for m in range(len(queue)-1,-1,-1):#m表示序号
                    if queue[m][1]==pan:
                        queue.insert(m+1,[order[1],groups[order[1]]])
                        flag=1
                        break
                if flag==0:
                    queue.append([order[1],groups[order[1]]])

        if order[0]=="DEQUEUE" and queue:
            print(queue.pop(0)[0])
    except EOFError:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

27928: 遍历树

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

思路：要判断哪个编号对应根节点，也就是不隶属于某一个节点的节点，分别把每组输入的第一项和其余项放在fangtou与exist里，对exist利用ass6中的去重办法

a=list(dict.fromkeys(a))，再逐一比较即可。

然后把每一棵树存在字典里便于读取时的索引

读取的递归程序有点像层次遍历

代码

```
#
class tree:
    def __init__(self,value):
        self.value=value
        self.children=[value]
n=int(input())
groups={}#用来放树
exist=[]#用来放所有树的孩子，然后去重
data=[]#用来存输入
fangtou=[]#把树根的value都放进去
#先收集数据
for i in range(n):
    a=[int(x) for x in input().split()]
    data.append(a)
#先找根对应的value
for m in data:#先找到根节点的value
    fangtou.append(m[0])
```

```
    exist.extend(m[1::])
exist=list(dict.fromkeys(exist))#去重

root_value=0
for p in fangtou:
    if p not in exist:
        root_value=p#即为根节点的value
        break
#然后对数据建树处理,groups只放各个树

for i in data:
    node=tree(i[0])
    node.children.extend(i[1::])#加进去的都是数字
    groups[str(i[0])]=node

root=groups[str(root_value)]


def pri(root):
    if len(root.children)==1:
        return root.children
    root.children.sort()
    stack=[]
    for i in root.children:
        if i==root.value:
            stack.append(i)
        else:
            stack.extend(pri(groups[str(i)]))

    return stack

a=pri(root)
for m in a:
    print(m)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==

#44521699提交状态

查看 提交 统计 提问

状态：Accepted

源代码

```
class tree:
    def __init__(self,value):
        self.value=value
        self.children=[value]
n=int(input())
groups={}#用来放树
exist=[]#用来放所有树的孩子，然后去重
data=[]#用来存输入
fangtou=[]#把树根的value都放进去
#先收集数据
for i in range(n):
    a=[int(x) for x in input().split()]
    data.append(a)
#先找根对应的value
for m in data:#先找到根节点的value
    fangtou.append(m[0])
    exist.extend(m[1::])
exist=list(dict.fromkeys(exist))#去重

root_value=0
for p in fangtou:
    if p not in exist:
        root_value=p#即为根节点的value
        break
#然后对数据建树处理,groups只放各个树

for i in data:
    node=tree(i[0])
```

基本信息

#: 44521699
题目: 27928
提交人: 23n2300013289
内存: 3904kB
时间: 27ms
语言: Python3
提交时间: 2024-04-03 20:25:50

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。==

很常规，ac了5道，本来也就抱着ac4去的，还算满足了，但是最后一道思路也很清晰，就是实现上可能有点繁琐没有时间了（我菜），吃完饭回来也ac了。庆幸的是周末可以有自由时间来解决上周的avl和欠了好久的每日选做了，激动！

因为想借这次机会模拟看看期末到底能ac几道（真的害怕只能做出1、2道）决定要不要中期退课，想看看自己的理想状态，就没去机房，结果还行（虽然老师放水，方法卡的很松吧？），如果在机房肯定还会受到破烂键盘和大佬同桌的心理压力的影响，（碎碎念：一定要克服一定要克服，全神贯注做好自己这112分钟）。

这次比较神奇的是，可能是平常手画树的递归画多了，很自然而然地就凭感觉写出来了两个递归程序，真的是巨大的进步，仍然在努力克服递归的恐惧。说明跟着闫老师真的学到了东西，这也是当初冒险想选这个班的初心，no pains，no gains。

暴露出来的问题是，除了树好像感觉不到什么新的知识点，基本都在用计概的方法实现，可能是因为思维量小的缘故，没有办法去联系大佬们说的堆、双端队列的方法，期末肯定不会这样简单，还要多加练习，多投入时间，才能学好。

至于昨天的笔试，感觉难度有限？如果有一定的记忆与对各种概念的确切理解是能做好的。

提醒自己要做：抓好比如搜索树，堆啊之类的概念，抓好各种树的节点数与层的关系，大致明白什么希尔排序、直接选择排序、冒泡排序，记住他们的时间复杂度，理解“稳定”的意义。