Assignment #7: April 月考

Updated 1557 GMT+8 Apr 3, 2024

2024 spring, Complied by 周添 物理学院

1. 题目

27706: 逐词倒放

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

代码

```
line = [i for i in input().split()]
line = reversed(line)
for j in line:
    print(j, end=' ')
```

状态: Accepted

源代码

```
line = [i for i in input().split()]
line = reversed(line)
for j in line:
    print(j, end=' ')
```

基本信息

#: 44560707
题目: 27706
提交人: 23n2300011538
内存: 3592kB
时间: 28ms
语言: Python3
提交时间: 2024-04-07 13:31:54

English 帮助 关于

27951: 机器翻译

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

代码

```
import queue

m, n = map(int, input().split())
nums = [int(i) for i in input().split()]
q = queue.Queue()
counter = 0
for i in nums:
    if i not in q.queue:
        counter += 1
        if q.qsize() == m:
            q.get()
        q.put(i)
```

```
print(counter)
```

27932: Less or Equal

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

代码

```
import heapq

m, n = map(int, input().split())
nums = [int(i) for i in input().split()]
heapq.heapify(nums)
if n == 0:
    k = heapq.heappop(nums)-1
    if k <= 0:
        print(-1)
    else:
        print(k)
else:
    for i in range(n - 1):
        heapq.heappop(nums)
    a = heapq.heappop(nums)
    if n == m:
        print(a)
    else:
        b = heapq.heappop(nums)
        if a < b:
            print(a)
        else:
            print(-1)
```

状态: Accepted

源代码

```
import heapq

m, n = map(int, input().split())
nums = [int(i) for i in input().split()]
heapq.heapify(nums)
if n == 0:
    k = heapq.heappop(nums)-1
    if k <= 0:
        print(-1)
    else:
        print(k)
else:
    for i in range(n - 1):
        heapq.heappop(nums)
    a = heapq.heappop(nums)
    if n == m:
        print(a)
    else:
        b = heapq.heappop(nums)
        if a < b:
            print(a)
        else:
            print(-1)
```

基本信息

#:	44560978
题目:	27932
提交人:	23n2300011538
内存:	10412kB
时间:	55ms
语言:	Python3
提交时间:	2024-04-07 14:12:30

27948: FBI树

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

代码

```
class Node:
    def __init__(self, v):
        self.v = v
        self.left = None
        self.right = None


def f_or_b_or_i(string):
    if string == '1'*len(string):
        return 'I'
    elif string == '0'*len(string):
        return 'B'
    else:
        return 'F'


def build_tree(s):
    if len(s) == 1:
        return Node(f_or_b_or_i(s))
    node = Node(f_or_b_or_i(s))
    mid = len(s)//2
    node.left = build_tree(s[:mid])
    node.right = build_tree(s[mid:])
    return node


def postorder_traversal(root):
```

```
    if root is None:
        return []

    left = postorder_traversal(root.left)
    right = postorder_traversal(root.right)

    return left + right + [root.v]


a = int(input())
st = input()
root = build_tree(st)
y = postorder_traversal(root)
print(''.join(y))
```

状态: Accepted

源代码

```
class Node:
    def __init__(self, v):
        self.v = v
        self.left = None
        self.right = None


def f_or_b_or_i(string):
    if string == '1'*len(string):
        return 'I'
    elif string == '0'*len(string):
        return 'B'
    else:
        return 'F'


def build_tree(s):
    if len(s) == 1:
        return Node(f_or_b_or_i(s))
    node = Node(f_or_b_or_i(s))
    mid = len(s)//2
    node.left = build_tree(s[:mid])
    node.right = build_tree(s[mid:])
    return node


def postorder_traversal(root):
    if root is None:
        return []
```

基本信息

#: 44561873
题目: 27948
提交人: 23n2300011538
内存: 3932kB
时间: 25ms
语言: Python3
提交时间: 2024-04-07 15:14:23

27925: 小组队列

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

代码

```
from collections import deque
t=int(input())
groups=dict()
for i in range(t):
    new_g=list(map(int,input().split()))
    for member in new_g:
        groups[member]=i
```

```
in_queue=deque()
seq=dict()
while True:
    order=input()
    if order=='STOP':
        break
    else:
        if order[0]=='E':
            new=int(order.split()[1])
            group=groups[new]
            if group in in_queue:
                seq[group].append(new)
            else:
                in_queue.append(group)
                seq[group]=deque()
                seq[group].append(new)

        else:
            head=in_queue[0]
            print(seq[head].popleft())
            if len(seq[head])==0:
                seq.pop(head)
                in_queue.popleft()
```

源代码

```
from collections import deque
t=int(input())
groups=dict()
for i in range(t):
    new_g=list(map(int,input().split()))
    for member in new_g:
        groups[member]=i
in_queue=deque()
seq=dict()
while True:
    order=input()
    if order=='STOP':
        break
    else:
        if order[0]=='E':
            new=int(order.split()[1])
            group=groups[new]
            if group in in_queue:
                seq[group].append(new)
            else:
                in_queue.append(group)
                seq[group]=deque()
                seq[group].append(new)

        else:
            head=in_queue[0]
            print(seq[head].popleft())
            if len(seq[head])==0:
                seq.pop(head)
```

基本信息

#:	44578994
题目:	27925
提交人:	23n2300011538
内存:	4912kB
时间:	109ms
语言:	Python3
提交时间:	2024-04-08 22:38:07

27928: 遍历树

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

代码

```
import heapq
class Node:
    def __init__(self,value):
        self.value=value
        self.child=[]
    def __lt__(self,other):
        return self.value<other.value
def build_tree(all_info,i):
    if len(all_info[i])==0:
        return Node(i)
    root=Node(i)
    for child in all_info[i]:
        root.child.append(build_tree(all_info,child))
    return root

def traversal(root):
    if len(root.child)==0:
        print(root.value)
        return
    childs=root.child
    heapq.heapify(childs)
    flag=0
    while childs:
        min_child=heapq.heappop(childs)
        #print(min_child.value)
        if flag==0:
            if root.value<min_child.value:
                print(root.value)
                flag=1
                traversal(min_child)
            else:
                traversal(min_child)
        else:
            traversal(min_child)
    if flag==0:
        print(root.value)
n=int(input())
all_info=dict()
for _ in range(n):
    info=list(map(int,input().split()))
    root=info[0]
    all_info[root]=info[1:]
keys=all_info.keys()
not_root=set()
for i in keys:
    not_root.update(all_info[i])
```

```
for i in keys:
    if i not in not_root:
        root_num=i
        break
#print(all_info)
root=build_tree(all_info,root_num)
traversal(root)
```

状态: Accepted

源代码

```
import heapq
class Node:
    def __init__(self,value):
        self.value=value
        self.child=[]
    def __lt__(self,other):
        return self.value<other.value
def build_tree(all_info,i):
    if len(all_info[i])==0:
        return Node(i)
    root=Node(i)
    for child in all_info[i]:
        root.child.append(build_tree(all_info,child))
    return root

def traversal(root):
    if len(root.child)==0:
        print(root.value)
        return
    childs=root.child
    heapq.heapify(childs)
    flag=0
    while childs:
        min_child=heapq.heappop(childs)
        #print(min_child.value)
        if flag==0:
            if root.value<min_child.value:
                print(root.value)
```

基本信息

#: 44579048
题目: 27928
提交人: 23n2300011538
内存: 3756kB
时间: 26ms
语言: Python3
提交时间: 2024-04-08 22:43:44

2. 学习总结和收获

前面的有点简单，最后两题不太会