Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Complied by 同学的姓名、院系

说明：

1）The complete process to learn DSA from scratch can be broken into 4 parts:

- Learn about Time and Space complexities
- Learn the basics of individual Data Structures
- Learn the basics of Algorithms
- Practice Problems on DSA

2）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。

3）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。

4）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 Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

1. 题目

02945: 拦截导弹

http://cs101.openjudge.cn/practice/02945/

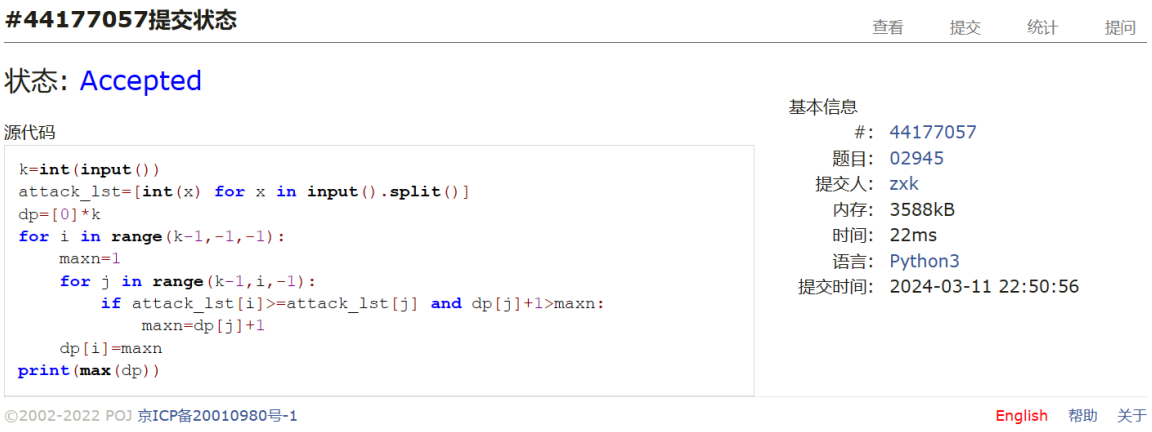
思路：

创建dp数组记录每个导弹位置的最大拦截数量，从最后一个向前遍历，在每个导弹，再次遍历到本位，如果后边的高度比前一个小，那就加一个拦截数量，之后把这个值赋给dp表示当前位置最大拦截数量，然后输出dp中的最大值

代码

```
k=int(input())
attack_lst=[int(x) for x in input().split()]
dp=[0]*k
for i in range(k-1,-1,-1):
    maxn=1
    for j in range(k-1,i,-1):
        if attack_lst[i]>=attack_lst[j] and dp[j]+1>maxn:
            maxn=dp[j]+1
    dp[i]=maxn
print(max(dp))
```

代码运行截图 （至少包含有"Accepted"）



04147:汉诺塔问题(Tower of Hanoi)

http://cs101.openjudge.cn/practice/04147

思路:

递归，从1挪x个到3，需要从1挪x-1个到2，然后挪1个到3，然后从2挪x-1个到3；为记录挪动的编号可采用栈模拟移动过程

代码

```
ipt_lst=[str(x) for x in input().split()]
n,a,b,c=int(ipt_lst[0]),ipt_lst[1],ipt_lst[2],ipt_lst[3]
move_lst=[]
a_stack=[a]
for i in range(n):
    a_stack.append(n-i)
b_stack=[b]
c_stack=[c]
def move(x,from_tower,with_tower,to_tower):
    if x>1:
        move(x-1,from_tower,to_tower,with_tower)
        move(1,from_tower,with_tower,to_tower)
```

```
        move(x-1,with_tower,from_tower,to_tower)
    else:
        action(from_tower,to_tower)
def action(ft,tt):
    tt.append(ft[-1])
    ft.pop()
    y=tt[-1]
    ans=str(y)+':'+ft[0]+'->'+tt[0]
    move_lst.append(ans)
move(n,a_stack,b_stack,c_stack)
for _ in move_lst:
    print(_)
```

代码运行截图 （至少包含有"Accepted"）

#44169729提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
ipt_lst=[str(x) for x in input().split()]
n,a,b,c=int(ipt_lst[0]),ipt_lst[1],ipt_lst[2],ipt_lst[3]
move_lst=[]
a_stack=[a]
for i in range(n):
    a_stack.append(n-i)
b_stack=[b]
c_stack=[c]
def move(x,from_tower,with_tower,to_tower):
    if x>1:
        move(x-1,from_tower,to_tower,with_tower)
        move(1,from_tower,with_tower,to_tower)
        move(x-1,with_tower,from_tower,to_tower)
    else:
        action(from_tower,to_tower)
def action(ft,tt):
    tt.append(ft[-1])
    ft.pop()
    y=tt[-1]
    ans=str(y)+':'+ft[0]+'->'+tt[0]
    move_lst.append(ans)
move(n,a_stack,b_stack,c_stack)
for _ in move_lst:
    print(_)
```

基本信息

#:	44169729
题目:	04147
提交人:	zxk
内存:	3596kB
时间:	22ms
语言:	Python3
提交时间:	2024-03-11 16:21:39

English 帮助 关于

03253: 约瑟夫问题No.2

http://cs101.openjudge.cn/practice/03253

思路:

首先输入每组的三个参数，然后对于每组，创建一个栈，对于孩子的循环，只要保证每个操作都是去除第一个，并将其补到最后一个即可，除非第一个是出局的人。首先操作p次，然后开始循环，同时循环和入栈，当栈的长度到达m时，出局一人，栈清空，如此循环，直到只剩一个人。

代码

```
casess=[]
while True:
    npmlst=[int(x) for x in input().split()]
    n,p,m=npmlst[0],npmlst[1],npmlst[2]
    if n==0 and p==0 and m ==0:
        break
    else:
        casess.append([n, p, m])
for case in casess:
    n,p,m=case[0],case[1],case[2]
    out_lst=[]
    stack=[]
    childrens=[int(x) for x in range(1,n+1)]
    for i in range(1,p):
        childrens.append(i)
        childrens.pop(0)
    while childrens:
        stack.append(childrens[0])
        childrens.append(childrens[0])
        childrens.pop(0)
        if len(stack)==m:
            out_lst.append(str(stack[-1]))
            stack.pop()
            childrens.pop()
            stack=[]
    ans=','.join(out_lst)
    print(ans)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
casess=[]
while True:
    npmlst=[int(x) for x in input().split()]
    n,p,m=npmlst[0],npmlst[1],npmlst[2]
    if n==0 and p==0 and m ==0:
        break
    else:
        casess.append([n, p, m])
for case in casess:
    n,p,m=case[0],case[1],case[2]
    out_lst=[]
    stack=[]
    childrens=[int(x) for x in range(1,n+1)]
    for i in range(1,p):
        childrens.append(i)
        childrens.pop(0)
    while childrens:
        stack.append(childrens[0])
        childrens.append(childrens[0])
        childrens.pop(0)
        if len(stack)==m:
            out_lst.append(str(stack[-1]))
            stack.pop()
            childrens.pop()
            stack=[]
    ans=','.join(out_lst)
    print(ans)
```

基本信息

#:	44090583
题目:	M03253
提交人:	zxk
内存:	3668kB
时间:	21ms
语言:	Python3
提交时间:	2024-03-06 15:48:58

21554:排队做实验 (greedy)v0.2

http://cs101.openjudge.cn/practice/21554

思路：

首先统计每个学生的时间，按每个学生的时间从小到大排序，然后确定学生顺序。让时间短的优先实验，总体等待时间即为最短。统计总共的时间再算平均时间。

代码

```
n=int(input())
time_lst=[int(x) for x in input().split()]
time_dct={}
for i in range(n):
    time_dct[i+1]=time_lst[i]
stu_time_lst=sorted(time_dct.items(),key=lambda x:x[1],reverse=False)
stu_lst=[]
for i in range(n):
    stu_lst.append(str(stu_time_lst[i][0]))
ave_wait=0
for i in range(n-1):
    ave_wait+=float((n-1-i)*stu_time_lst[i][1]/n)
print(' '.join(stu_lst))
print('%.2f'%ave_wait)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44093395提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=int(input())
time_lst=[int(x) for x in input().split()]
time_dct={}
for i in range(n):
    time_dct[i+1]=time_lst[i]
stu_time_lst=sorted(time_dct.items(),key=lambda x:x[1],reverse=False)
stu_lst=[]
for i in range(n):
    stu_lst.append(str(stu_time_lst[i][0]))
ave_wait=0
for i in range(n-1):
    ave_wait+=float((n-1-i)*stu_time_lst[i][1]/n)
print(' '.join(stu_lst))
print('%.2f'%ave_wait)
```

基本信息

#: 44093395
题目: 21554
提交人: zxk
内存: 3644kB
时间: 21ms
语言: Python3
提交时间: 2024-03-06 17:13:58

English 帮助 关于

19963:买学区房

http://cs101.openjudge.cn/practice/19963

思路：首先把二维位置改写为距离，然后按照房子创建两个字典，为性价比和开销，然后对性价比和开销进行排序，找到各自的中位数，最后统计符合要求的房子数

代码

```
n=int(input())
dst_dct={}
pairs = [i[1:-1] for i in input().split()]
dst_lst = [sum(map(int,i.split(','))) for i in pairs]
for i in range(n):
    dst_dct[i]=dst_lst[i]
cost_lst=[int(x) for x in input().split()]
cost_dct={}
for i in range(n):
    cost_dct[i]=cost_lst[i]
value_dct={}
value_lst=[]
for i in range(n):
    value=float(dst_lst[i]/cost_lst[i])
    value_dct[i]=value
    value_lst.append(value)
value_lst.sort()
cost_lst.sort()
if n%2==0:
    mid_cost=0.5*(cost_lst[int(n/2)-1]+cost_lst[int(n/2)])
    mid_value=0.5*(value_lst[int(n/2)-1]+value_lst[int(n/2)])
else:
    mid_cost=cost_lst[int(n/2)]
    mid_value=value_lst[int(n/2)]
H=0
for i in range(n):
    if value_dct[i]>mid_value and cost_dct[i]<mid_cost:
        H+=1
print(H)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

#44092141提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=int(input())
dst_dct={}
pairs = [i[1:-1] for i in input().split()]
dst_lst = [sum(map(int,i.split(','))) for i in pairs]
for i in range(n):
    dst_dct[i]=dst_lst[i]
cost_lst=[int(x) for x in input().split()]
cost_dct={}
for i in range(n):
    cost_dct[i]=cost_lst[i]
value_dct={}
value_lst=[]
for i in range(n):
    value=float(dst_lst[i]/cost_lst[i])
    value_dct[i]=value
    value_lst.append(value)
value_lst.sort()
cost_lst.sort()
if n%2==0:
    mid_cost=0.5*(cost_lst[int(n/2)-1]+cost_lst[int(n/2)])
    mid_value=0.5*(value_lst[int(n/2)-1]+value_lst[int(n/2)])
else:
    mid_cost=cost_lst[int(n/2)]
    mid_value=value_lst[int(n/2)]
H=0
for i in range(n):
    if value_dct[i]>mid_value and cost_dct[i]<mid_cost:
        H+=1
print(H)
```

基本信息

#: 44092141
题目: T19963
提交人: zxk
内存: 4936kB
时间: 25ms
语言: Python3
提交时间: 2024-03-06 16:41:31

English 帮助 关于

27300: 模型整理

http://cs101.openjudge.cn/practice/27300

思路:

把待整理的模型按照名称分割，入字典，对字典里的每个item分析，value是列表，把列表先换算为M，排序之后再换回去，然后按照key直接排序即可，最后输出字符串

代码

```
n=int(input())#n个待整理的模型
model_lst=[]
for _ in range(n):
    model_lst.append(input().split("-"))
#n个元素的列表，每个元素是['名称','参数量']
model_dct={}
for i in range(n):
    model=model_lst[i]
    name,value=model[0],list(model[1])
    if value[-1]=='M':
        value.pop()
        value=float(''.join(value))
    else:
        value.pop()
        value=float(''.join(value))*1000
    if name in model_dct.keys():
        model_dct[name]=model_dct[name]+[value]
    else:
```

```
        model_dct[name]=[value]
#名称为key的字典，item为列表，包含参数量浮点数
ans_dct={}
for name in model_dct.keys():
    name_lst=model_dct[name]
    name_lst.sort(reverse=False)
    for i in range(len(name_lst)):
        if name_lst[i]<=999:
            if name_lst[i]%1==0:
                name_lst[i]=int(name_lst[i])
            name_lst[i]=str(name_lst[i])+'M'
        else:
            num=float(name_lst[i]/1000)
            if num%1==0:
                num=int(num)
            name_lst[i]=str(num)+'B'
    ans_dct[name]=name_lst
ans_dct=dict(sorted(ans_dct.items()))
for name in ans_dct.keys():
    print(name+': '+', '.join(ans_dct[name]))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

状态: Accepted

源代码

```
n=int(input())#n个待整理的模型
model_lst=[]
for _ in range(n):
    model_lst.append(input().split("-"))
#n个元素的列表，每个元素是['名称','参数量']
model_dct={}
for i in range(n):
    model=model_lst[i]
    name,value=model[0],list(model[1])
    if value[-1]=='M':
        value.pop()
        value=float(''.join(value))
    else:
        value.pop()
        value=float(''.join(value))*1000
    if name in model_dct.keys():
        model_dct[name]=model_dct[name]+[value]
    else:
        model_dct[name]=[value]
#名称为key的字典，item为列表，包含参数量浮点数
ans_dct={}
for name in model_dct.keys():
    name_lst=model_dct[name]
    name_lst.sort(reverse=False)
    for i in range(len(name_lst)):
        if name_lst[i]<=999:
            if name_lst[i]%1==0:
                name_lst[i]=int(name_lst[i])
            name_lst[i]=str(name_lst[i])+'M'
        else:
            num=float(name_lst[i]/1000)
            if num%1==0:
                num=int(num)
            name_lst[i]=str(num)+'B'
    ans_dct[name]=name_lst
ans_dct=dict(sorted(ans_dct.items()))
for name in ans_dct.keys():
    print(name+': '+', '.join(ans_dct[name]))
```

基本信息

#:	44167521
题目:	27300
提交人:	zxk
内存:	3704kB
时间:	21ms
语言:	Python3
提交时间:	2024-03-11 11:41:57

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。

月考时间里AC2，有很大原因是算法想出来但是在语法上不太会实现，然后卡了很长时间。

只把最简单的几个做出了，拦截导弹的读题有问题，导致试了很多次也没对，浪费时间。

1.拦截导弹：

完全不会dp或dfs，学了一下dp，看了一下答案，之后会找一些dp的题做

2.汉诺塔问题：

递归确实掌握的不好，需要注意递归所满足的三个条件，思路不是很好想，另外需要学会用栈模拟移动过程，完成记录。

3.约瑟夫问题：

模拟运行是最普遍的思路，不过里边有一个循环的实现，只要把第一个去掉，最后一个补上就能实现循环，直到列表为空。

4.排队做实验：

思路很简单，主要是对于字典排序的语法掌握不太熟，lambda语法没有太理解。

5.买学区房：

字典真神

6.模型整理：

字典真神+1，字典排序可以直接按照字典序排序，非常之方便啊