

Assignment #3: March月考

Updated 1537 GMT+8 March 6, 2024

2024 spring, Compiled by ==狄晨阳 生命科学学院==

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: Windows11

Python编程环境: Spyder IDE 5.4.3

C/C++编程环境: 无

1. 题目

02945: 拦截导弹

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

思路：类似动态规划的思路，从最后向最前一个个检索，比较正常上来的值和所有比它小的高度上来的值中最大值加一的大小，取最大值作为其值，再输出其中的最大值

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Thu Mar  7 23:36:31 2024
```

```
@author: 20311
"""

k=int(input())
missles=list(map(int,input().split()))
missles.reverse()
a=[1]*k
for j in range(k-1):
    if missles[j+1]>=missles[j]:
        a[j+1]+=a[j]
    for l in range(j+1):
        if missles[l]<=missles[j+1]:
            a[j+1]=max(a[j+1],a[l]+1)
print(max(a))
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 7 23:36:31 2024

@author: 20311
"""

k=int(input())
missles=list(map(int,input().split()))
missles.reverse()
a=[1]*k
for j in range(k-1):
    if missles[j+1]>=missles[j]:
        a[j+1]+=a[j]
    for l in range(j+1):
        if missles[l]<=missles[j+1]:
            a[j+1]=max(a[j+1],a[l]+1)
print(max(a))
```

基本信息

#: 44112521
 题目: 02945
 提交人: 23n2300012138(yukino)
 内存: 3612kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-08 00:23:45

04147:汉诺塔问题(Tower of Hanoi)

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

思路：一开始没有想清楚，后来在纸上打了一下草稿，通过递归实现了，即将n从起点到终点，先需将n-1从起点到起点终点之外的另一个点，如此反复即可

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Fri Mar 8 23:59:31 2024

@author: 20311
"""

data=input().split()
```

```

n=int(data[0])
d=data[1:4]
dd={1:2,3:0,2:1}
def move(n,a,b):
    if n==1:
        print('{}: {}->{}'.format(1,d[a],d[b]))
    else:
        for i in range(n-1,-1,-1):
            move(i,a,dd[a+b])
            print('{}: {}->{}'.format(i+1,d[a],d[b]))
            a=dd[a+b]
        return
move(n,0,2)

```

代码运行截图 == (至少包含有"Accepted") ==

状态: **Accepted**

源代码

```

# -*- coding: utf-8 -*-
"""
Created on Fri Mar 8 23:59:31 2024

@author: 20311
"""

data=input().split()
n=int(data[0])
d=data[1:4]
dd={1:2,3:0,2:1}
def move(n,a,b):
    if n==1:
        print('{}: {}->{}'.format(1,d[a],d[b]))
    else:
        for i in range(n-1,-1,-1):
            move(i,a,dd[a+b])
            print('{}: {}->{}'.format(i+1,d[a],d[b]))
            a=dd[a+b]
        return
move(n,0,2)

```

基本信息

#: 44125690
 题目: 04147
 提交人: 23n2300012138(yukino)
 内存: 3592kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-09 03:09:44

03253: 约瑟夫问题No.2

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

思路：通过程序模拟报数的过程，用另一个列表储存答案

代码

```

# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 15:48:03 2024

@author: 20311
"""

```

```

while True:
    n,p,m=map(int,input().split())
    if n==0 and p==0 and m==0:
        break
    else:
        circle=[x for x in range(1,n+1)]
        ans=[]
        p-=1
        while len(circle)>0:
            p-=1
            p=(p+m)%n
            ans.append(circle[p])
            del circle[p]
            n-=1
        print(' '.join(map(str,ans)))

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```

# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 15:48:03 2024

@author: 20311
"""

while True:
    n,p,m=map(int,input().split())
    if n==0 and p==0 and m==0:
        break
    else:
        circle=[x for x in range(1,n+1)]
        ans=[]
        p-=1
        while len(circle)>0:
            p-=1
            p=(p+m)%n
            ans.append(circle[p])
            del circle[p]
            n-=1
        print(' '.join(map(str,ans)))

```

基本信息

#: 44154587
 题目: 03253
 提交人: 23n2300012138(yukino)
 内存: 3608kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-10 15:58:37

21554:排队做实验 (greedy)v0.2

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

思路：读取数据后用字典套列表储存对应序号以处理时间相同时的先后问题，然后将时间按短到长排序，并储存其序号并输出，最后计算平均等待时间并输出

代码

```

#
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 16:12:30 2024

```

```

@author: 20311
"""

n=int(input())
stu=list(map(int,input().split()))
d={}
for i in range(n):
    if stu[i] not in d:
        d[stu[i]]=[i+1]
    else:
        d[stu[i]].append(i+1)
stu.sort()
order=[]
ss=-1
t=0
for s in stu:
    if ss!=s:
        t=0
        order.append(str(d[s][t]))
    else:
        t+=1
        order.append(str(d[s][t]))
    ss=s
print(' '.join(order))
ans=0
for i in range(1,n):
    ans+=i*stu[-i-1]
print('{:.2f}'.format(ans/n))

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 16:12:30 2024

@author: 20311
"""

n=int(input())
stu=list(map(int,input().split()))
d={}
for i in range(n):
    if stu[i] not in d:
        d[stu[i]]=[i+1]
    else:
        d[stu[i]].append(i+1)
stu.sort()
order=[]
ss=-1
t=0
for s in stu:
    if ss!=s:
        t=0
        order.append(str(d[s][t]))
    else:
        t+=1
        order.append(str(d[s][t]))
    ss=s
print(' '.join(order))
ans=0
for i in range(1,n):
    ans+=i*stu[-i-1]
print('{:.2f}'.format(ans/n))
```

基本信息

#: 44155694
题目: 21554
提交人: 23n2300012138(yukino)
内存: 3672kB
时间: 21ms
语言: Python3
提交时间: 2024-03-10 16:29:01

19963:买学区房

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

思路：读取数据后计算出性价比和价格的中位数，然后比较是否有符合条件的房子，有几个并输出

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 16:47:55 2024

@author: 20311
"""

n=int(input())
p=[i[1:-1] for i in input().split()]
d=[sum(map(int,x.split(','))) for x in p]
prices=list(map(int,input().split()))
xjb=[d[i]/prices[i] for i in range(n)]

xjb1=xjb.copy()
prices1=prices.copy()
xjb1.sort()
prices1.sort()
if n%2==0:
    mid_xjb=(xjb1[n//2-1]+xjb1[n//2])/2
```

```

        mid_prices=(prices1[n//2-1]+prices1[n//2])/2
    else:
        mid_xjb=xjb1[(n+1)//2-1]
        mid_prices=prices1[(n+1)//2-1]
    c=0
    for i in range(n):
        if xjb[i]>mid_xjb and prices[i]<mid_prices:
            c+=1
    print(c)

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: **Accepted**

源代码

```

# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 16:47:55 2024

@author: 20311
"""

n=int(input())
p=[i[1:-1] for i in input().split()]
d=[sum(map(int,x.split(','))) for x in p]
prices=list(map(int,input().split()))
xjb=[d[i]/prices[i] for i in range(n)]

xjb1=xjb.copy()
prices1=prices.copy()
xjb1.sort()
prices1.sort()
if n%2==0:
    mid_xjb=(xjb1[n//2-1]+xjb1[n//2])/2
    mid_prices=(prices1[n//2-1]+prices1[n//2])/2
else:
    mid_xjb=xjb1[(n+1)//2-1]
    mid_prices=prices1[(n+1)//2-1]
c=0
for i in range(n):
    if xjb[i]>mid_xjb and prices[i]<mid_prices:
        c+=1
print(c)

```

基本信息

#: 44156904
 题目: 19963
 提交人: 23n2300012138(yukino)
 内存: 4288kB
 时间: 26ms
 语言: Python3
 提交时间: 2024-03-10 16:57:27

27300: 模型整理

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

思路: 先读取数据, 然后对其用“-”分割, 前半部分储存为name便于排序, 并用字典将前半部分与后半部分对应分类, 然后对字典中每个名称下的数据量大小通过自定义的函数进行排序, 最后对name排序, 并输出分类后的结果

代码

```

# # -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 17:02:38 2024

@author: 20311

```

```

"""
def change(x):
    a=float(x[:-1])
    if x[-1]=='B':
        return a*1000
    else:
        return a

n=int(input())
models=[]
for _ in range(n):
    models.append(input())
name=[]
d={}
for m in models:
    a,b=m.split('-')
    if a not in d:
        d[a]=[b]
        name.append(a)
    else:
        d[a].append(b)

for na in name:
    d[na].sort(key=lambda x:change(x))
name.sort()
for na in name:
    s=', '.join(d[na])
    print('{}: {}'.format(na,s))

```

代码运行截图 == (AC代码截图, 至少包含有"Accepted") ==

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 17:02:38 2024

@author: 20311
"""
def change(x):
    a=float(x[:-1])
    if x[-1]=='B':
        return a*1000
    else:
        return a

n=int(input())
models=[]
for _ in range(n):
    models.append(input())
name=[]
d={}
for m in models:
    a,b=m.split('-')
    if a not in d:
        d[a]=[b]
        name.append(a)
    else:
        d[a].append(b)

for na in name:
    d[na].sort(key=lambda x:change(x))
name.sort()
for na in name:
    s=' '.join(d[na])
    print('{}: {}'.format(na,s))
```

基本信息

#: 44157362
题目: 27300
提交人: 23n2300012138(yukino)
内存: 3644kB
时间: 22ms
语言: Python3
提交时间: 2024-03-10 17:13:52

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。==

有

前两道题花的时间相对多了一点，主要还是一开始思路没有理清楚，凭着一点感觉就写了，后来好好理清楚之后再通过代码实现就快了许多。以后还是要想清楚再开始写，不然就是浪费时间得不偿失。然后也是类似的题要多做一些，做过了之后后面就会好很多。