

Assignment #2: 编程练习

2024 spring, Compiled by 胡豪俊 工学院

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) 课程网站是Canvas平台, <https://pku.instructure.com>, 学校通知3月1日导入选课名单后启用。**作业写好后，保留在自己手中，待3月1日提交。**

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

操作系统: Windows11

Python编程环境: Visual Studio Code

1. 题目

27653: Fraction类

http://cs101.openjudge.cn/2024sp_routine/27653/

代码

```
def gcd(m,n):
    while m&n!=0:
        x=m%n
        m=n
        n=x
    return n

class Fraction:
    def __init__(self,top,bottom):
        self.num=top
        self.den=bottom
```

```
def __str__(self):
    return(str(self.num)+"/"+str(self.den))

def __add__(self, otherfraction):
    newnum=self.num*otherfraction.den+self.den*otherfraction.num
    newden=self.den*otherfraction.den
    common=gcd(self.den, otherfraction.den)
    return(Fraction(newnum//common, newden//common))
```

```
L=[int(i) for i in input().split()]
print(Fraction(L[0],L[1])+Fraction(L[2],L[3]))
```

##我感觉我还是不太清楚类想做什么，大概理解了一下，一个类可以看作一个集合，里面对特定的元素进行考虑，类中有一系列方法，在涉及类中对象的操作时会调用这些方法，目前我只能理解到这里

##这题没有什么难度，用欧几里得算法就能轻松解决

#44076903提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
def gcd(m,n):
    while m&n!=0:
        x=m%n
        m=n
        n=x
    return n

class Fraction:
    def __init__(self,top,bottom):
        self.num=top
        self.den=bottom

    def __str__(self):
        return(str(self.num)+"/"+str(self.den))

    def __add__(self, otherfraction):
        newnum=self.num*otherfraction.den+self.den*otherfraction.num
        newden=self.den*otherfraction.den
        common=gcd(self.den, otherfraction.den)
        return(Fraction(newnum//common, newden//common))

L=[int(i) for i in input().split()]
print(Fraction(L[0],L[1])+Fraction(L[2],L[3]))
```

基本信息

#: 44076903
 题目: 27653
 提交人: 22n2200011800
 内存: 3652kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-05 15:54:16

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

04110: 圣诞老人的礼物-Santa Clau's Gifts

greedy/dp, <http://cs101.openjudge.cn/practice/04110>

代码

```
n,m=input().split()
n=int(n)
m=int(m)
present={}
k=[]
i=1
j=0
while i<=n:
    w=[int(j) for j in input().split()]
    x=w[0]/w[1]
    k.append([x,w[1]])
```

```

    i+=1
k.sort(reverse=True)
kilo=0
value=0
while kilo<=m and j<=n-1:
    if kilo+k[j][1]<=m:
        value+=k[j][0]*k[j][1]
        kilo+=k[j][1]
    else:
        value+=(m-kilo)*k[j][0]
        break
    j+=1
print(round(value,1))

```

##这题思路比较清晰，比第一次做好多了，第一次做死活没反应过来可以相同的平均价值，错用字典，注意到之后就比较简单了

#44077687提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

n,m=input().split()
n=int(n)
m=int(m)
present={}
k=[]
i=1
j=0
while i<=n:
    w=[int(j) for j in input().split()]
    x=w[0]/w[1]
    k.append([x,w[1]])
    i+=1
k.sort(reverse=True)
kilo=0
value=0
while kilo<=m and j<=n-1:
    if kilo+k[j][1]<=m:
        value+=k[j][0]*k[j][1]
        kilo+=k[j][1]
    else:
        value+=(m-kilo)*k[j][0]
        break
    j+=1
print(round(value,1))

```

基本信息

#: 44077687
 题目: 04110
 提交人: 22n2200011800
 内存: 3672kB
 时间: 22ms
 语言: Python3
 提交时间: 2024-03-05 16:26:19

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

18182: 打怪兽

implementation/sortings/data structures, <http://cs101.openjudge.cn/practice/18182/>

代码

```

n=int(input())
i=1
ans=[]
while i<=n:
    L=[int(j) for j in input().split()]
    j=1
    skill={}
    hurt={}

    while j<=L[0]:
        z=[int(k) for k in input().split()]

```

```

        if z[0] in skill.keys():
            skill[z[0]]+=z[1]
        else:
            skill[z[0]]=z[1]
        j+=1
    r=list(skill.keys())
    r.sort()
    for x in r:
        a=skill[x]
        a.sort(reverse=True)
        if len(a)<=L[1]:
            hurt[x]=sum(a)
        else:
            hurt[x]=sum(a[:L[1]])
    total=0
    for x in r:
        total+=hurt[x]
        if total>=L[2]:
            ans.append(str(x))
            break
    if total<L[2]:
        ans.append("alive")
    i+=1
for t in ans:
    print(t)

```

##伤害最大化打满一套，如果在途中死了就跳出记录时间，活着就**alive**

##需要注意的是打满一套有诸多限制，比如每秒出手次数，是否同一时间，以及各自出手的伤害，比较好的方法用把时间和伤害（列表存储）搞成字典，然后判断

状态: Accepted

源代码

```
n=int(input())
i=1
ans=[]
while i<=n:
    L=[int(j) for j in input().split()]
    j=1
    skill={}
    hurt={}

    while j<=L[0]:
        z=[int(k) for k in input().split()]
        if z[0] in skill.keys():
            skill[z[0]]+=z[1]
        else:
            skill[z[0]]=z[1]
        j+=1
    r=list(skill.keys())
    r.sort()
    for x in r:
        a=skill[x]
        a.sort(reverse=True)
        if len(a)<=L[1]:
            hurt[x]=sum(a)
        else:
            hurt[x]=sum(a[:L[1]])

    total=0
    for x in r:
        total+=hurt[x]
        if total>=L[2]:
            ans.append(str(x))
            break
    if total<L[2]:
        ans.append("alive")
    i+=1
for t in ans:
    print(t)
```

截图(Alt + A)

基本信息

#: 44081693
题目: 18182
提交人: 22n2200011800
内存: 3824kB
时间: 81ms
语言: Python3
提交时间: 2024-03-05 20:17:20

230B. T-primes

binary search/implementation/math/number theory, 1300, <http://codeforces.com/problemset/problem/230/B>

代码

```
import math
def primejudge(num):
    if num<=1:
        return False
    if num==2:
        return True
    if num!=2 and num%2==0:
        return False
    for i in range(3, int(num**0.5) + 1, 2):
        if num % i == 0:
            return False
    return True

n=int(input())
NUM=[int(i) for i in input().split()]
k=[]
for t in NUM:
    x=math.sqrt(t)
    x=math.floor(x)
```

```

if x**2==t:
    k.append(x)
else:
    k.append(1)
for t in k:
    if t==1 or primejudge(t) is False:
        print("NO")
        continue
    else:
        print("YES")

```

##没能用python测试ac用了pypy，依稀记得这题第一次做的时候就做了很久，前前后后提交了四十多次才ac，这次也优化了比较多次，交上去之后重新去了解了一下筛法，现在会一点了
##本题我依然用了这个方法，在2050那题中就改用筛法了

249742827	Mar/05/2024 17:14	Accepted
---------------------------	----------------------	----------

1364A. XXXXX

brute force/data structures/number theory/two pointers, 1200, <https://codeforces.com/problemset/problem/1364/A>

代码

```

t=int(input())
ans=[]
for k in range(t):
    n,x=map(int,input().split())
    L=list(map(int,input().split()))
    m=-1
    s=sum(L)
    if s%x!=0:
        ans.append(n)
    else:
        for i in range(n):
            s-=L[i]
            if s%x!=0:
                m=max(i+1,n-i-1,m)##如果一边不整除，那么另一边也不整除
        ans.append(m)
for t in ans:
    print(t)

```

##这题卡了很久很久，因为每次都想着便利一点用sum函数就可以了，枚举出每个子序列然后取模，但是这样其实浪费了大量算力，tle得人都快疯了

##这题就属于是典型的思路谁都会，但是不深入思考肯定过不了的，在群里看到老师发的其他同学的证明对我启发很大，谢谢老师和同学！

Submission	Time	Verdict
250617861	Mar/11/2024 14:04	Accepted

18176: 2050年成绩计算

<http://cs101.openjudge.cn/practice/18176/>

代码

```
from math import sqrt
N=10010
s=[True]*N
p=2
while p*p<=N:
    if s[p] is True:
        for i in range(p*2,N,p):
            s[i]=False
    p+=1
s[0]=False
s[1]=False
m,n=map(int,input().split())
for i in range(m):
    L=list(map(int,input().split()))
    sum=0
    for num in L:
        x=int(sqrt(num))
        if s[x] is True and num==x*x:
            sum+=num
    average=sum/len(L)
    if sum==0:
        print(0)
    else:
        print('%.2f' % average)
##终于能用代码实现筛法了
```

#44173310提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
from math import sqrt
N=10010
s=[True]*N
p=2
while p*p<=N:
    if s[p] is True:
        for i in range(p*2,N,p):
            s[i]=False
    p+=1
s[0]=False
s[1]=False
m,n=map(int,input().split())
for i in range(m):
    L=list(map(int,input().split()))
    sum=0
    for num in L:
        x=int(sqrt(num))
        if s[x] is True and num==x*x:
            sum+=num
    average=sum/len(L)
    if sum==0:
        print(0)
    else:
        print('%.2f' % average)
```

基本信息

#: 44173310
题目: 18176
提交人: 22n2200011800
内存: 4204kB
时间: 60ms
语言: Python3
提交时间: 2024-03-11 19:41:42

2. 学习总结和收获

这周命途多舛，先是当天写代码在自己限制的时间内有两题总是tle，一怒之下先去写别的作业了，结果最后差点忘了，在ddl前才想起来。总的来说我花在数算上的时间太少，一些基本的数据结构和基本方法还没有很好的掌握，但是我每天都会关注群里大家讨论的内容，老师和同学对题目的很多看法给了我巨大无比的帮助，希望能越学越好。