Assignment #2: 编程练习

Updated 0953 GMT+8 Feb 24, 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）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

==（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

27653: Fraction类

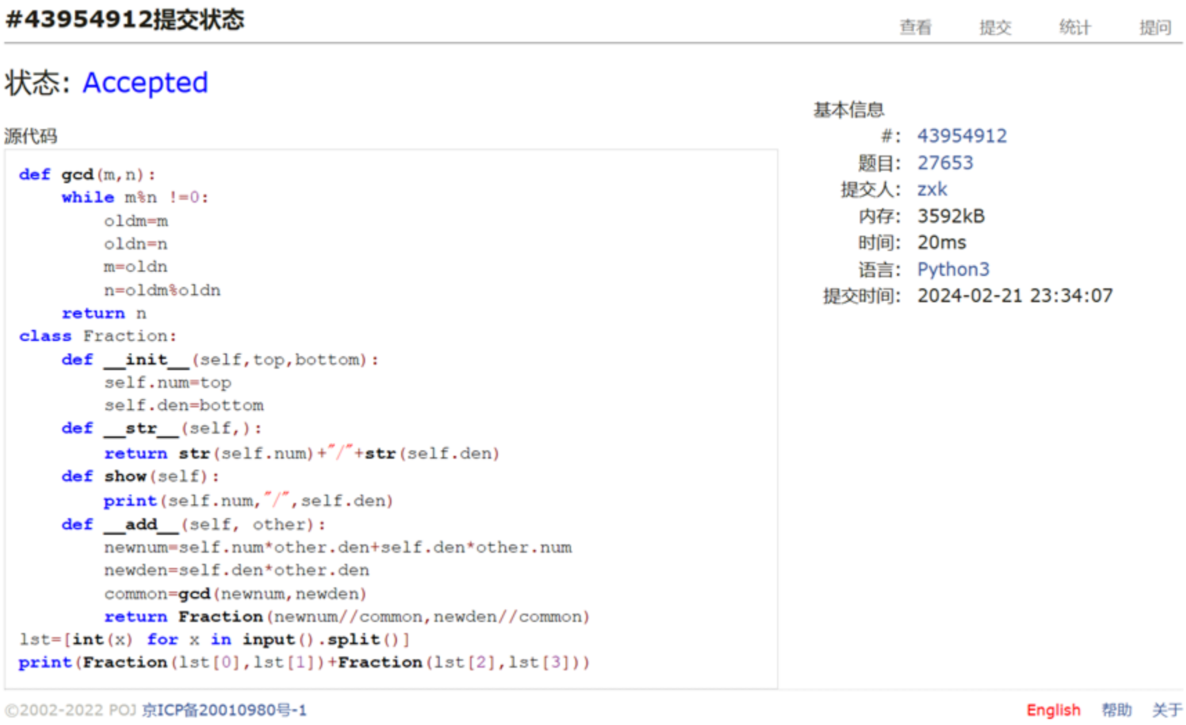
http://cs101.openjudge.cn/2024sp_routine/27653/

思路：

代码

```
def gcd(m,n):
    while m%n !=0:
        oldm=m
        oldn=n
        m=oldn
        n=oldm%oldn
    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 show(self):
        print(self.num,"/",self.den)
    def __add__(self, other):
        newnum=self.num*other.den+self.den*other.num
        newden=self.den*other.den
        common=gcd(newnum,newden)
        return Fraction(newnum//common,newden//common)
lst=[int(x) for x in input().split()]
print(Fraction(lst[0],lst[1])+Fraction(lst[2],lst[3]))
```

代码运行截图 == （至少包含有"Accepted"） ==



#43954912提交状态

状态: Accepted

源代码

```
def gcd(m,n):
    while m%n !=0:
        oldm=m
        oldn=n
        m=oldn
        n=oldm%oldn
    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 show(self):
        print(self.num,"/",self.den)
    def __add__(self, other):
        newnum=self.num*other.den+self.den*other.num
        newden=self.den*other.den
        common=gcd(newnum,newden)
        return Fraction(newnum//common,newden//common)
lst=[int(x) for x in input().split()]
print(Fraction(lst[0],lst[1])+Fraction(lst[2],lst[3]))
```

基本信息

#:	43954912
题目:	27653
提交人:	zxk
内存:	3592kB
时间:	20ms
语言:	Python3
提交时间:	2024-02-21 23:34:07

©2002-2022 POJ 京ICP备20010980号-1

04110: 圣诞老人的礼物-Santa Clau’s Gifts

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

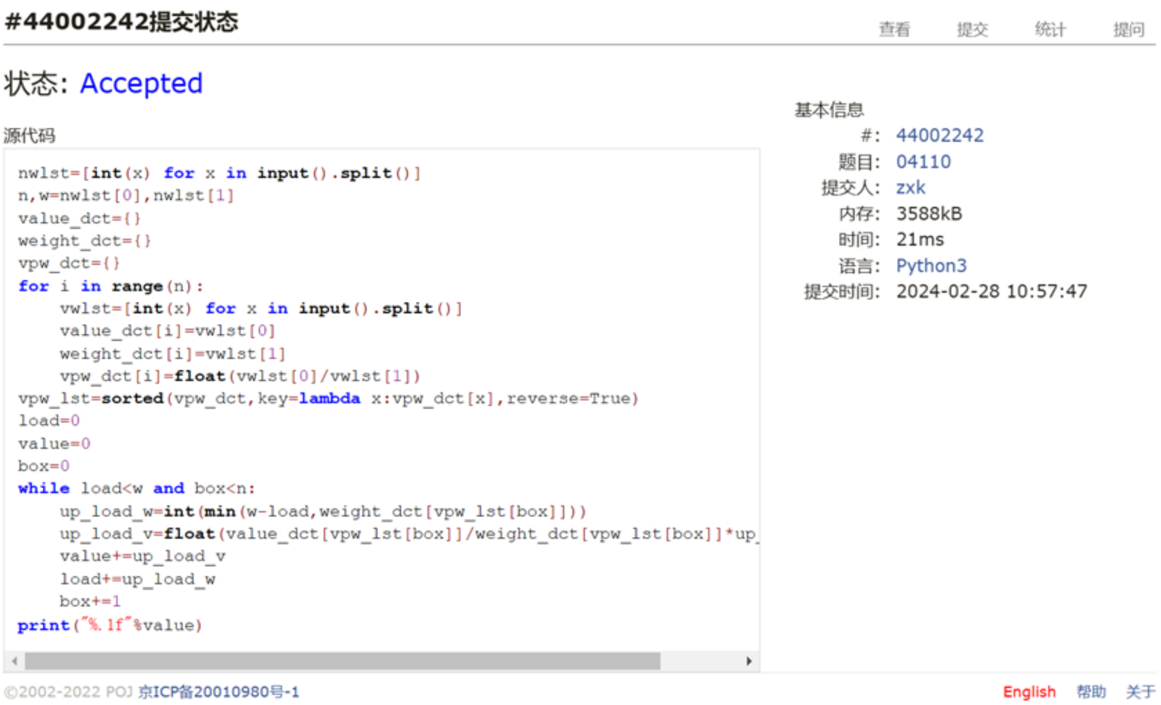
思路：

分别做价值、重量、单价的字典，用箱子的序号作为统一的key，把单价字典按items排序，在负载未满且盒子并非最后一个时，从单价高到低上载到盒子空了或负载满了，更新负载价值，负载和盒子序号，输出负载

代码

```
nwlst=[int(x) for x in input().split()]
n,w=nwlst[0],nwlst[1]
value_dct={}
weight_dct={}
vpw_dct={}
for i in range(n):
    vwlst=[int(x) for x in input().split()]
    value_dct[i]=vwlst[0]
    weight_dct[i]=vwlst[1]
    vpw_dct[i]=float(vwlst[0]/vwlst[1])
vpw_lst=sorted(vpw_dct,key=lambda x:vpw_dct[x],reverse=True)
load=0
value=0
box=0
while load<w and box<n:
    up_load_w=int(min(w-load,weight_dct[vpw_lst[box]]))
    up_load_v=float(value_dct[vpw_lst[box]]/weight_dct[vpw_lst[box]]*up_load_w)
    value+=up_load_v
    load+=up_load_w
    box+=1
print("%.1f"%value)
```

代码运行截图 ==（至少包含有"Accepted"）==



#44002242提交状态

状态: Accepted

源代码

```
nwlst=[int(x) for x in input().split()]
n,w=nwlst[0],nwlst[1]
value_dct={}
weight_dct={}
vpw_dct={}
for i in range(n):
    vwlst=[int(x) for x in input().split()]
    value_dct[i]=vwlst[0]
    weight_dct[i]=vwlst[1]
    vpw_dct[i]=float(vwlst[0]/vwlst[1])
vpw_lst=sorted(vpw_dct,key=lambda x:vpw_dct[x],reverse=True)
load=0
value=0
box=0
while load<w and box<n:
    up_load_w=int(min(w-load,weight_dct[vpw_lst[box]]))
    up_load_v=float(value_dct[vpw_lst[box]]/weight_dct[vpw_lst[box]]*up_
    value+=up_load_v
    load+=up_load_w
    box+=1
print("%.1f"%value)
```

基本信息

#: 44002242
题目: 04110
提交人: zxk
内存: 3588kB
时间: 21ms
语言: Python3
提交时间: 2024-02-28 10:57:47

©2002-2022 POJ 京ICP备20010980号-1

18182: 打怪兽

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

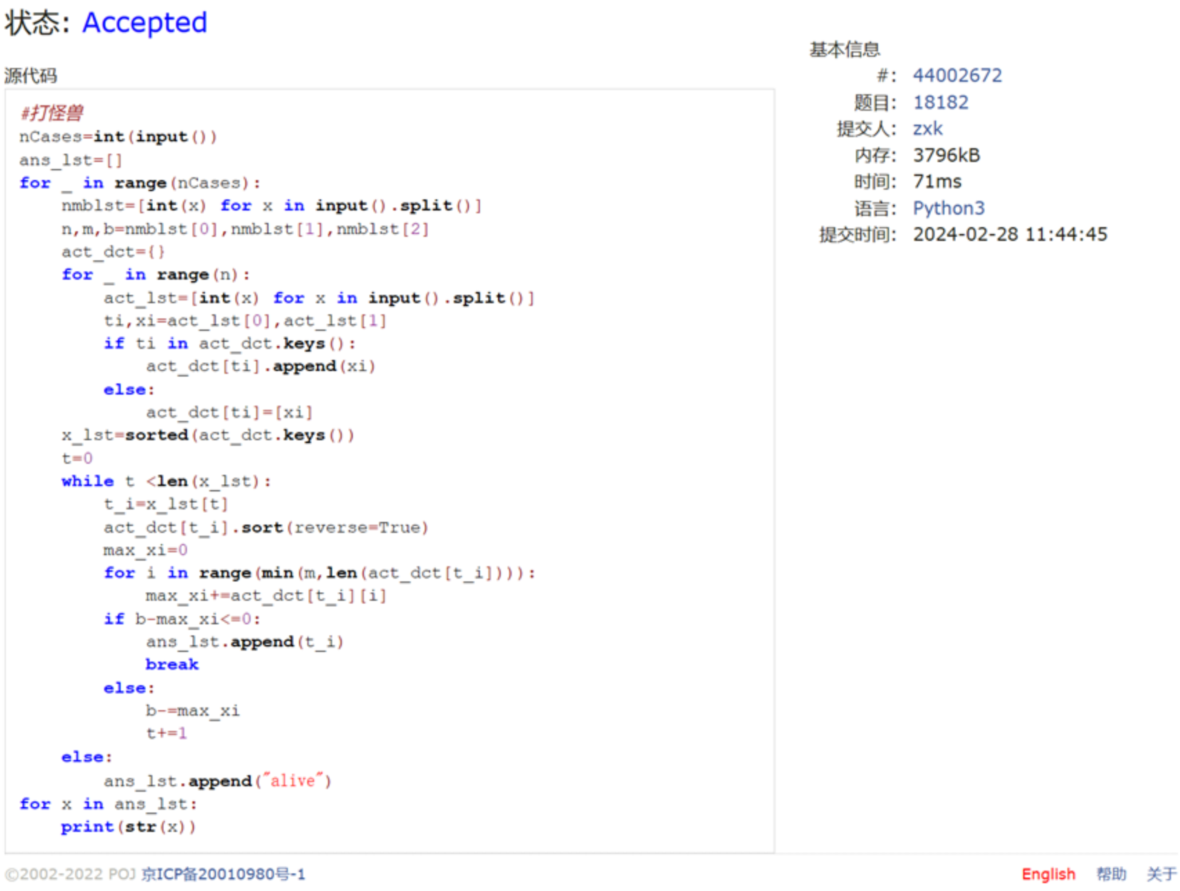
思路:

对于每个情况：首先创造行动字典，key是回合序号，item是不同技能的伤害列表，若ti在keys中就append，否则添加字典元素并放入第一个值，这时我们得到一个完整的行动字典，对keys整理得到顺序列表，设定指针t为当前进行的回合，当回合小于最多进行的回合数时，按从高到低排列行动字典中这个回合的技能伤害，计算出本轮能打出的最大伤害（最多释放的技能数和本轮拥有的技能数中的较小值），这时判断是否已经打死，如果打死了，就把此时的回合序号添加到输出列表，否则怪物的血量下降，进入下一回合，直到回合用尽，怪物存活，输出alive

代码

```
nCases=int(input())
ans_lst=[]
for _ in range(nCases):
    nmblst=[int(x) for x in input().split()]
    n,m,b=nmblst[0],nmblst[1],nmblst[2]
    act_dct={}
    for _ in range(n):
        act_lst=[int(x) for x in input().split()]
        ti,xi=act_lst[0],act_lst[1]
        if ti in act_dct.keys():
            act_dct[ti].append(xi)
        else:
            act_dct[ti]=[xi]
    x_lst=sorted(act_dct.keys())
    t=0
    while t <len(x_lst):
        t_i=x_lst[t]
        act_dct[t_i].sort(reverse=True)
        max_xi=0
        for i in range(min(m,len(act_dct[t_i]))):
            max_xi+=act_dct[t_i][i]
        if b-max_xi<=0:
            ans_lst.append(t_i)
            break
        else:
            b-=max_xi
            t+=1
    else:
        ans_lst.append("alive")
for x in ans_lst:
    print(str(x))
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```
#打怪兽
nCases=int(input())
ans_lst=[]
for _ in range(nCases):
    nmblst=[int(x) for x in input().split()]
    n,m,b=nmblst[0],nmblst[1],nmblst[2]
    act_dct={}
    for _ in range(n):
        act_lst=[int(x) for x in input().split()]
        ti,xi=act_lst[0],act_lst[1]
        if ti in act_dct.keys():
            act_dct[ti].append(xi)
        else:
            act_dct[ti]=[xi]
    x_lst=sorted(act_dct.keys())
    t=0
    while t <len(x_lst):
        t_i=x_lst[t]
        act_dct[t_i].sort(reverse=True)
        max_xi=0
        for i in range(min(m,len(act_dct[t_i]))):
            max_xi+=act_dct[t_i][i]
        if b-max_xi<=0:
            ans_lst.append(t_i)
            break
        else:
            b-=max_xi
            t+=1
    else:
        ans_lst.append("alive")
for x in ans_lst:
    print(str(x))
```

基本信息

#:	44002672
题目:	18182
提交人:	zxk
内存:	3796kB
时间:	71ms
语言:	Python3
提交时间:	2024-02-28 11:44:45

230B. T-primes

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

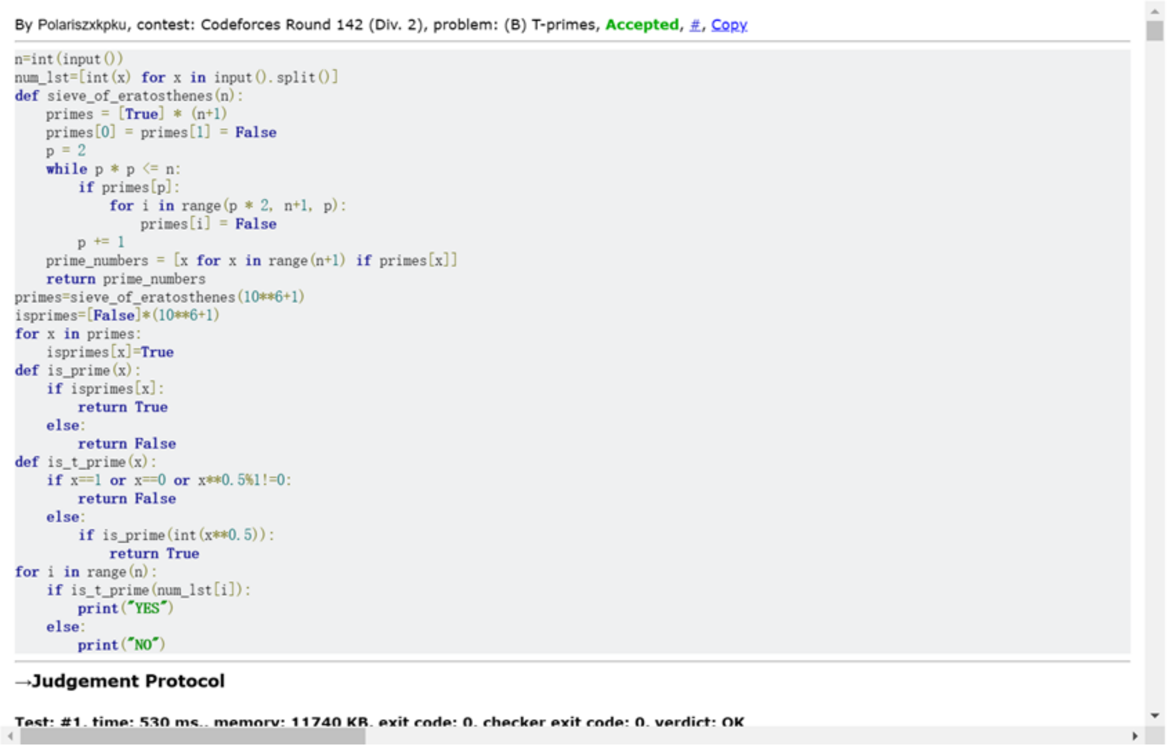
思路：

T质数实际就是首先保证时平方数，然后根是质数，问题出现在时间限制，需要采用埃氏筛

代码

```python
n=int(input())
num_lst=[int(x) for x in input().split()]
def sieve_of_eratosthenes(n):
    primes = [True] * (n+1)
    primes[0] = primes[1] = False
    p = 2
    while p * p <= n:
        if primes[p]:
            for i in range(p * 2, n+1, p):
                primes[i] = False
        p += 1
    prime_numbers = [x for x in range(n+1) if primes[x]]
    return prime_numbers
primes=sieve_of_eratosthenes(10**6+1)
isprimes=[False]*(10**6+1)
for x in primes:
    isprimes[x]=True
def is_prime(x):
    if isprimes[x]:
        return True
    else:
        return False
def is_t_prime(x):
    if x==1 or x==0 or x**0.5%1!=0:
        return False
    else:
        if is_prime(int(x**0.5)):
            return True
for i in range(n):
    if is_t_prime(num_lst[i]):
        print("YES")
    else:
        print("NO")
```

代码运行截图 == （AC代码截图，至少包含有"Accepted"） ==



By Polariszxkpku, contest: Codeforces Round 142 (Div. 2), problem: (B) T-primes, Accepted, #, Copy

```python
n=int(input())
num_lst=[int(x) for x in input().split()]
def sieve_of_eratosthenes(n):
    primes = [True] * (n+1)
    primes[0] = primes[1] = False
    p = 2
    while p * p <= n:
        if primes[p]:
            for i in range(p * 2, n+1, p):
                primes[i] = False
        p += 1
    prime_numbers = [x for x in range(n+1) if primes[x]]
    return prime_numbers
primes=sieve_of_eratosthenes(10**6+1)
isprimes=[False]*(10**6+1)
for x in primes:
    isprimes[x]=True
def is_prime(x):
    if isprimes[x]:
        return True
    else:
        return False
def is_t_prime(x):
    if x==1 or x==0 or x**0.5%1!=0:
        return False
    else:
        if is_prime(int(x**0.5)):
            return True
for i in range(n):
    if is_t_prime(num_lst[i]):
        print("YES")
    else:
        print("NO")
```

→Judgement Protocol

Test: #1, time: 530 ms., memory: 11740 KB, exit code: 0, checker exit code: 0, verdict: OK

1364A. XXXXX

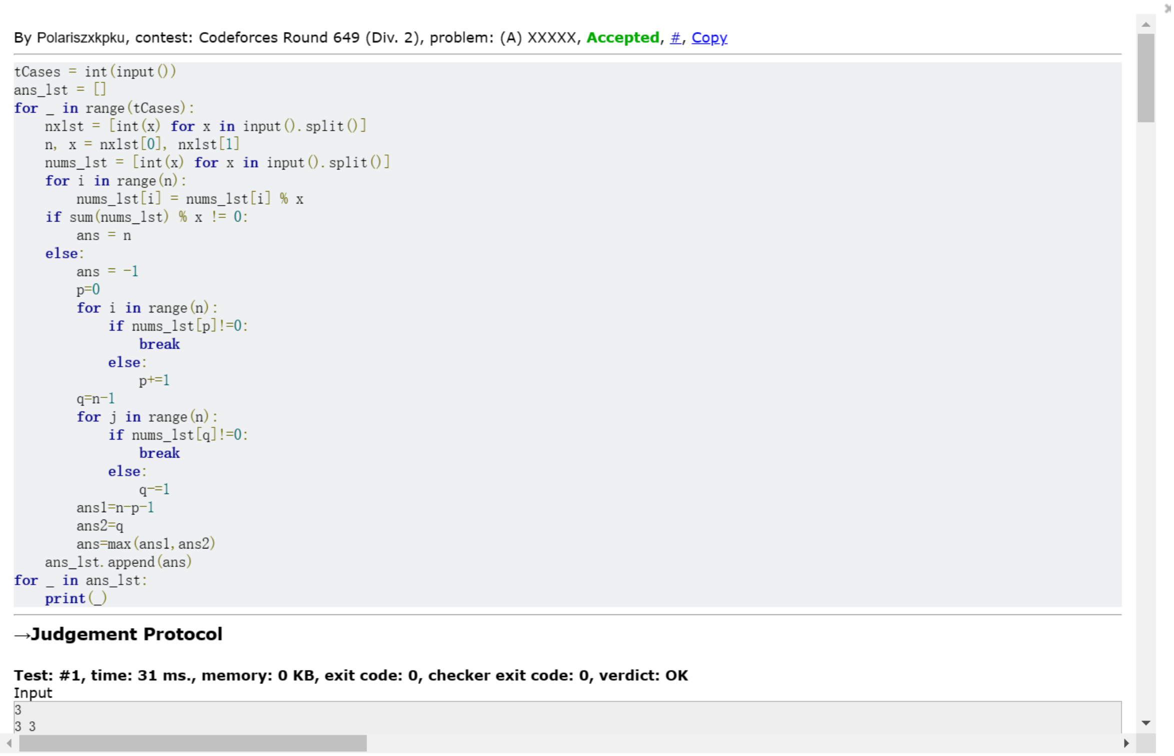
brute force/data structures/number theory/two pointers, 1200, https://codeforces.com/problemset/problem/1364/A

思路：先取余数，从左到右找，直到第一个非x倍数，从右往左找同理，然后取更长的

代码

```python
tCases = int(input())
ans_lst = []
for _ in range(tCases):
    nxlst = [int(x) for x in input().split()]
    n, x = nxlst[0], nxlst[1]
    nums_lst = [int(x) for x in input().split()]
    for i in range(n):
        nums_lst[i] = nums_lst[i] % x
    if sum(nums_lst) % x != 0:
        ans = n
    else:
        ans = -1
        p=0
        for i in range(n):
            if nums_lst[p]!=0:
                break
            else:
                p+=1
        q=n-1
        for j in range(n):
            if nums_lst[q]!=0:
                break
            else:
                q-=1
        ans1=n-p-1
        ans2=q
        ans=max(ans1,ans2)
    ans_lst.append(ans)
for _ in ans_lst:
    print(_)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



By Polariszxkpku, contest: Codeforces Round 649 (Div. 2), problem: (A) XXXXX, **Accepted**, #, Copy

```
tCases = int(input())
ans_lst = []
for _ in range(tCases):
    nxlst = [int(x) for x in input().split()]
    n, x = nxlst[0], nxlst[1]
    nums_lst = [int(x) for x in input().split()]
    for i in range(n):
        nums_lst[i] = nums_lst[i] % x
    if sum(nums_lst) % x != 0:
        ans = n
    else:
        ans = -1
        p=0
        for i in range(n):
            if nums_lst[p]!=0:
                break
            else:
                p+=1
        q=n-1
        for j in range(n):
            if nums_lst[q]!=0:
                break
            else:
                q-=1
        ans1=n-p-1
        ans2=q
        ans=max(ans1,ans2)
    ans_lst.append(ans)
for _ in ans_lst:
    print(_)
```

→Judgement Protocol

Test: #1, time: 31 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```
3
3 3
```

18176: 2050年成绩计算

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

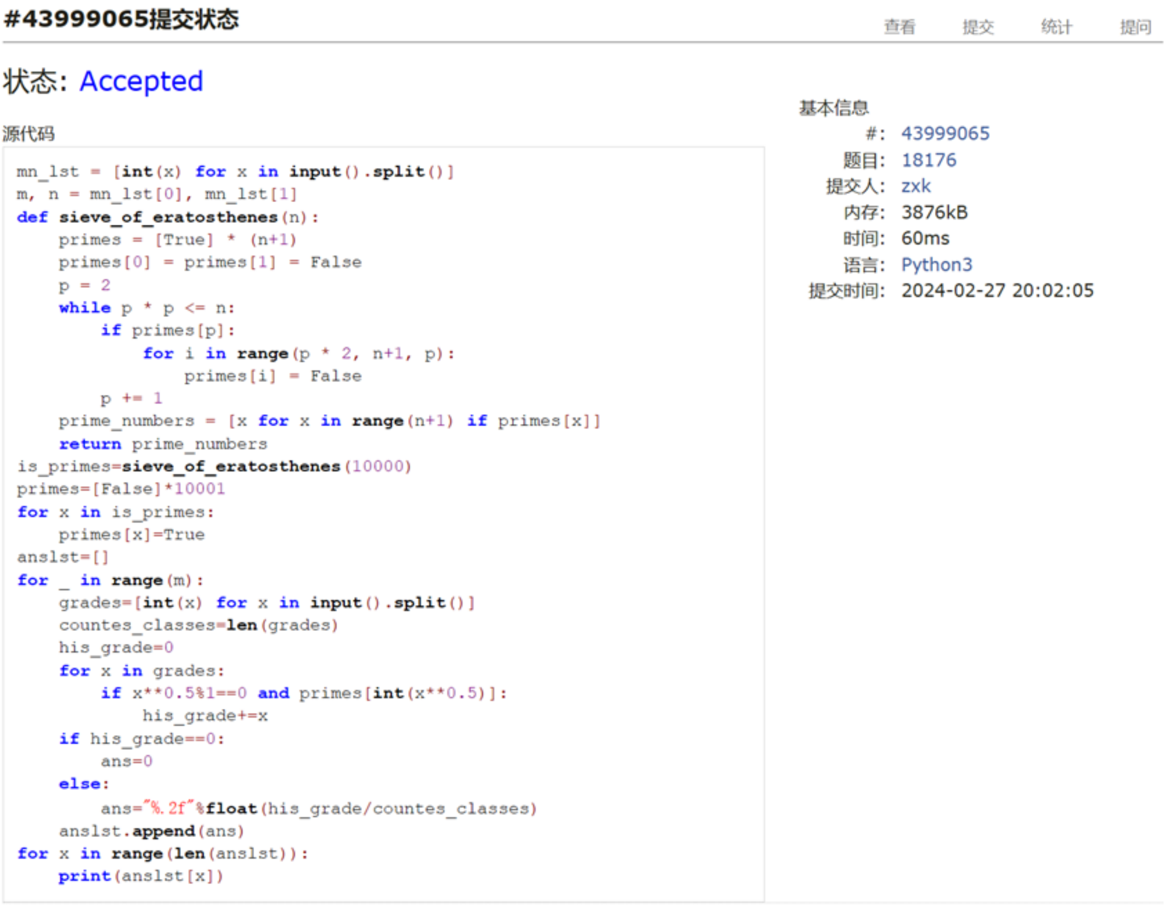
思路：

同T素数

代码

```
mn_lst = [int(x) for x in input().split()]
m, n = mn_lst[0], mn_lst[1]
def sieve_of_eratosthenes(n):
    primes = [True] * (n+1)
    primes[0] = primes[1] = False
    p = 2
    while p * p <= n:
        if primes[p]:
            for i in range(p * 2, n+1, p):
                primes[i] = False
        p += 1
    prime_numbers = [x for x in range(n+1) if primes[x]]
    return prime_numbers
is_primes=sieve_of_eratosthenes(10000)
primes=[False]*10001
for x in is_primes:
    primes[x]=True
anslst=[]
for _ in range(m):
    grades=[int(x) for x in input().split()]
    countes_classes=len(grades)
    his_grade=0
    for x in grades:
        if x**0.5%1==0 and primes[int(x**0.5)]:
            his_grade+=x
    if his_grade==0:
        ans=0
    else:
        ans="%.2f"%float(his_grade/countes_classes)
    anslst.append(ans)
for x in range(len(anslst)):
    print(anslst[x])
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#43999065提交状态

状态: Accepted

源代码

```
mn_lst = [int(x) for x in input().split()]
m, n = mn_lst[0], mn_lst[1]
def sieve_of_eratosthenes(n):
    primes = [True] * (n+1)
    primes[0] = primes[1] = False
    p = 2
    while p * p <= n:
        if primes[p]:
            for i in range(p * 2, n+1, p):
                primes[i] = False
        p += 1
    prime_numbers = [x for x in range(n+1) if primes[x]]
    return prime_numbers
is_primes=sieve_of_eratosthenes(10000)
primes=[False]*10001
for x in is_primes:
    primes[x]=True
anslst=[]
for _ in range(m):
    grades=[int(x) for x in input().split()]
    countes_classes=len(grades)
    his_grade=0
    for x in grades:
        if x**0.5%1==0 and primes[int(x**0.5)]:
            his_grade+=x
    if his_grade==0:
        ans=0
    else:
        ans="%.2f"%float(his_grade/countes_classes)
    anslst.append(ans)
for x in range(len(anslst)):
    print(anslst[x])
```

基本信息

#: 43999065
题目: 18176
提交人: zxk
内存: 3876kB
时间: 60ms
语言: Python3
提交时间: 2024-02-27 20:02:05

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。==

1.Fraction类的约分过程、埃氏筛并不熟悉，需要放cheating paper上；

2.对“class”的用法还没完全会，可以再练习练习；

3.问题比较复杂的时候，命名最好可以采用更具有实际意义的名字，虽然可能比较冗长，但是代码写出来是给人看的，乱套了再debug也太痛苦了；

4.打怪兽这个题感觉就是对现实中判断游戏进程的模拟，思路不难但是考虑的细节很多；

5.查找列表中是否含有这个元素比给出列表中的第几个元素耗时多得多，不如打出素数表然后转化为一系列True/False的表，再找对应为止的元素

6.xxx的题对于算法要求很重要，试了很多方法，受困于语法的实现，wa了很多次，才想明白怎么算最合理