

# Assignment #4: T-primes + 贪心

Updated 0337 GMT+8 Oct 15, 2024

2024 fall, Compiled by <mark>同学的姓名、院系</mark>

**姓名：李彦臻**

**学号：2300010821**

**学院：数学科学学院**

**\*\*说明：\*\***

1) 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。

## 1. 题目

### 34B. Sale

greedy, sorting, 900, <https://codeforces.com/problemset/problem/34/B>

思路：这道题从本质上来讲，只需要把商品按照价格从大到小排序并取前 b 项即可。但唯一的问题就是可能会出现负数，所以只需要把负数在第一步统全部统一为 0 即可～（这样相加的时候不会对结果产生任何影响）

代码

```
```python
#
```

number=list(map(int,input().split()))
a=number[0]
b=number[1]
```

```

price=list(map(int,input().split()))


for i in range(a):
    if price[i] > 0:
        price[i]=0#所有的正数变为 0（反正永远也不会买）
        #统一为 0 了之后，以后直接买价格前 b 个产品即可！方便~


price.sort()#从小到大排列
allow=price[0:b]#只保留前 b 个元素

n=-sum(allow)
print(n)

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>




  
Jo1423 | [Logout](#)

[HOME](#)
[TOP](#)
[CATALOG](#)
[CONTESTS](#)
[GYM](#)
[PROBLEMSET](#)
[GROUPS](#)
[RATING](#)
[EDU](#)
[API](#)
[CALENDAR](#)
[HELP](#)

[PROBLEMS](#)
[SUBMIT CODE](#)
[MY SUBMISSIONS](#)
[STATUS](#)
[HACKS](#)
[ROOM](#)
[STANDINGS](#)
[CUSTOM INVOCATION](#)

| #         | Author              | Problem                 | Lang     | Verdict  | Time   | Memory | Sent                   | Judged                 |   |                         |
|-----------|---------------------|-------------------------|----------|----------|--------|--------|------------------------|------------------------|---|-------------------------|
| 283495946 | Practice:<br>Jo1423 | <a href="#">34B</a> - 4 | Python 3 | Accepted | 154 ms | 20 KB  | 2024-09-29<br>11:01:50 | 2024-09-29<br>11:01:50 | ★ | <a href="#">Compare</a> |

[→ Source](#)
[Copy](#)

```

number=list(map(int,input().split()))
a=number[0]
b=number[1]

price=list(map(int,input().split()))

for i in range(a):
    if price[i] > 0:
        price[i]=0#所有的正数变为0（反正永远也不会买）
        #统一为0了之后，以后直接买价格前b个产品即可！方便~

price.sort()#从小到大排列
allow=price[0:b]#只保留前b个元素

n=-sum(allow)
print(n)

```

[Click to see test details](#)

### 160A. Twins

greedy, sortings, 900, <https://codeforces.com/problemset/problem/160/A>

思路：把硬币按照金额从大到小进行排序，所需要硬币的最少数量显然就是把硬币从最大的开始取一直到超过半数的时候的数量。为此，我们设计一个“变化的量”，对其不断累加硬币的金额，直到其金额超过半数时停止，并记录下此刻的硬币数量即可~

代码

```

python

m=int(input())#硬币个数
values=list(map(int,input().split()))


n=sum(values)#总金额数
values.sort(reverse=True)#从大到小排序

total=0#将 total 定义为一个随时改变的量，记录每时每刻的总和

for i in range(m):
    total += values[i]
    if total >= (n // 2 + 1):#临界值：n//2+1
        print(i + 1)#记得加一（列表计数规则的特殊性）
        break

```

代码运行截图 ==（至少包含有“Accepted”）==



[Jo1423](#) | [Logout](#)

[HOME](#)
[TOP](#)
[CATALOG](#)
[CONTESTS](#)
[GYM](#)
[PROBLEMS](#)
[SUBMIT CODE](#)
[MY SUBMISSIONS](#)
[STATUS](#)
[HACKS](#)
[ROOM](#)
[STANDINGS](#)
[CUSTOM INVOCATION](#)

| #         | Author              | Problem                   | Lang     | Verdict  | Time   | Memory | Sent                   | Judged                 |   |                         |
|-----------|---------------------|---------------------------|----------|----------|--------|--------|------------------------|------------------------|---|-------------------------|
| 283507004 | Practice:<br>Jo1423 | <a href="#">160A</a> - 20 | Python 3 | Accepted | 154 ms | 16 KB  | 2024-09-29<br>12:23:25 | 2024-09-29<br>12:23:25 | ★ | <a href="#">Compare</a> |

[Source](#)
[Copy](#)

```

m=int(input())#硬币个数
values=list(map(int,input().split()))

n=sum(values)#总金额数
values.sort(reverse=True)#从大到小排序

total=0#将total定义为一个随时改变的量，记录每时每刻的总和

for i in range(m):
    total += values[i]
    if total >= (n // 2 + 1):#临界值：n//2+1
        print(i + 1)#记得加一（列表计数规则的特殊性）
        break

```

[Click to see test details](#)

### 1879B. Chips on the Board

constructive algorithms, greedy, 900,  
<https://codeforces.com/problemset/problem/1879/B>

思路：对于这道题，不妨设每行每列的权重均单调递减（ $a_1 < a_2 < \dots$  且  $b_1 < b_2 < \dots$ ）；下证：答案就是  $n \cdot a_1 + \sigma_B$  和  $n \cdot b_1 + \sigma_A$  中的较小者！

原理：首先，不妨设  $a_i$  和  $b_i$  单调递增（平移行和列不影响结果）；易知，至少有  $n$  个 chip（否则一定有一个行和一个列一个 chip 都没有，交点便矛盾）

情况 1. 如果有一行一个 chip 都没有，则每一列都必须至少有一个 chip（否则交点矛盾）  
#此时，列和  $\geq \sigma_B$ ；而行和本来就  $\geq n \cdot a_1$ （因为至少有  $n$  个 chip，每个 chip 对应的行均  $\geq a_1$ ），  
故总和  $\geq n \cdot a_1 + \sigma_B$

情况 2. 如果有一列一个 chip 都没有，则每一行都必须至少有一个 chip（否则交点矛盾）  
#此时，行和  $\geq \sigma_A$ ；而列和本来就  $\geq n \cdot b_1$ （因为至少有  $n$  个 chip，每个 chip 对应的行均  $\geq b_1$ ），  
故总和  $\geq n \cdot b_1 + \sigma_A$

情况 3. 如果每行每列都有 chip，则行和  $\geq \sigma_A$ ，列和  $\geq \sigma_B$ ，此时总和  $\geq \sigma_A + \sigma_B$   
显然大于等于上面两种情况（舍去）

综上，由情况一和情况二可知证明完毕！

代码

```
```python

...

amount=int(input())
result=[]

for i in range(amount):
    n=int(input())
    line=list(map(int,input().split()))
    column=list(map(int,input().split()))
    line.sort()#将 ai 按大小顺序排列
    column.sort()
    A=n * line[0] + sum(column)#第一种情况
    B=n * column[0] + sum(line)#第二种情况
    result.append(min(A,B))

for results in result:
    print(results)
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

General										
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
283728245	Practice: Jo1423	1879B - 10	Python 3	Accepted	452 ms	50252 KB	2024-09-30 11:41:06	2024-09-30 11:41:06	☆	Compare

→ Source Copy

```
amount=int(input())
result=[]
for i in range(amount):
    n=int(input())
    line=list(map(int,input().split()))
    column=list(map(int,input().split()))
    line.sort()#将ai按大小顺序排列
    column.sort()
    A=n * line[0] + sum(column)#第一种情况
    B=n * column[0] + sum(line)#第二种情况
    result.append(min(A,B))
for results in result:
    print(results)
```

[Click to see test details](#)

### 158B. Taxi

\*special problem, greedy, implementation, 1100,  
<https://codeforces.com/problemset/problem/158/B>

思路：先把四人的小组扔进去，正好占据一个车；再把三人的小组扔进去，并用一人的小组填充空缺；再把两人的小组扔进去，并用一人的小组填充最后一辆车的空缺（此时要根据最后一辆车是否有空缺进行讨论）；最后再把剩下的一人小组扔进去！

代码

```
```python
```

```
```
```

```
n=int(input())
amount=list(map(int,input().split()))
a=amount.count(4)
b=amount.count(3)
c=amount.count(2)
d=amount.count(1)
```

k = a#先把四人的小组扔进去

k += b#再把三人的小组扔进去，并用一人的小组填充空缺


```



if d > b:
    d -= b
else:
    d=0
k += (c//2)#再把两人的小组扔进去，并用一人的小组填充最后一辆车的空缺
if c % 2 != 0:#最后一辆车有空缺
    k += 1
    if d > 2:
        d -= 2
    else:
        d=0
k += (d//4)#最后再把一人的小组扔进去
if d % 4 != 0:
    k += 1

print(k)

```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>



 | 
 
[Jo1423](#) | [Logout](#)

[HOME](#)
[TOP](#)
[CATALOG](#)
[CONTESTS](#)
[GYM](#)
[PROBLEMSET](#)
[GROUPS](#)
[RATING](#)
[EDU](#)
[API](#)
[CALENDAR](#)
[HELP](#)

[PROBLEMS](#)
[SUBMIT CODE](#)
[MY SUBMISSIONS](#)
[STATUS](#)
[STANDINGS](#)
[CUSTOM INVOCATION](#)

| #         | Author              | Problem                   | Lang     | Verdict  | Time   | Memory  | Sent                | Judged              |  |                         |
|-----------|---------------------|---------------------------|----------|----------|--------|---------|---------------------|---------------------|--|-------------------------|
| 286309838 | Practice:<br>Jo1423 | <a href="#">158B</a> - 10 | Python 3 | Accepted | 156 ms | 3252 KB | 2024-10-17 05:48:11 | 2024-10-17 05:48:11 |  | <a href="#">Compare</a> |

[Source](#)
[Copy](#)

```

n=int(input())
amount=list(map(int,input().split()))
a=amount.count(4)
b=amount.count(3)
c=amount.count(2)
d=amount.count(1)

k = a#先把四人的小组扔进去
k += b#再把三人的小组扔进去，并用一人的小组填充空缺
if d > b:
    d -= b
else:
    d=0
k += (c//2)#再把两人的小组扔进去，并用一人的小组填充最后一辆车的空缺
if c % 2 != 0:#最后一辆车有空缺
    k += 1
    if d > 2:
        d -= 2
    else:
        d=0
k += (d//4)#最后再把一人的小组扔进去
if d % 4 != 0:
    k += 1

print(k)

```

[Click](#) to see test details

### \*230B. T-primes（选做）

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

思路：使用正常的方法来检验一个数开根是否为素数，会因为  $n$  小于  $10^5$ ， $number$  小于  $10^{12}$ ，并且复杂度是： $o(\sqrt{number}) * (n)$ ，导致计算量最大可能会达到  $10^{11}$ ，远大于系统所允许的  $10^8$ ，因此要换一个方法！！

此时，注意到显然只有  $p^2$  合乎题意，而小于  $10^{12}$  的  $p^2$  中， $p$  必须小于  $10^6$ ！因此只需（使用欧拉筛的方法）找到所有小于  $10^6$  的素数，把他们平方并放进一个列表，然后再验证那些数在不在列表里即可！

必须要注意的是，在集合中检索元素的时间复杂度为  $o(1)$ ，所以在查找一个数是否为素数的过程中一定要把列表转化为集合！！

此时复杂度= $o(10^6) + o(10^6) + o(10^5) = o(10^6)$ ，可以运行~

代码

```
```python

```
n=10**6
primes=[]
status=['True']*(n + 1)#用欧拉筛的方法找素数，只用 o(n)！（原理见 py0.3）
for i in range(2, n + 1):#这一步复杂度=o(10^6)
    if status[i] == 'True':
        primes.append(i)
    for prime in primes:
        if prime*i > n:
            break
        else:
            status[prime*i]='False'
        if i % prime == 0:
            break
prime_squared=set()#注意！！在集合中检索元素的时间复杂度为 o(1)，所以要用集合！！
#但是，要小心：如果直接写 my_set={}, 系统会识别为字典！因此要写为=set()！！
for prime in primes:#这一步复杂度=o(10^6)
    prime_squared.add(prime**2)#集合没有 append，只有 add；且 add 不会造成重复~
result=[]
m=int(input())
numbers=list(map(int, input().split()))
for number in numbers:#这一步复杂度=o(10^5)
    if number in prime_squared:
        result.append('YES')
    else:
        result.append('NO')
for word in result:
    print(word)
```

代码运行截图 <mark>（至少包含有“Accepted”）</mark>

CODEFORCES

Sponsored by TON

Jo1423 | Logout

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

| #         | Author           | Problem                   | Lang     | Verdict  | Time    | Memory   | Sent                | Judged              |  |                         |
|-----------|------------------|---------------------------|----------|----------|---------|----------|---------------------|---------------------|--|-------------------------|
| 285794431 | Practice: Jo1423 | <a href="#">230B</a> - 28 | Python 3 | Accepted | 1562 ms | 29080 KB | 2024-10-14 12:05:55 | 2024-10-14 12:05:55 |  | <a href="#">Compare</a> |

→ Source

Copy

```
# 上个代码因为n小于10^5, number小于10^12, 而复杂度是: o(sqrt(number)*n),
# 因此计算量最大会达到10^11, 远大于10^8, 要换一个方法!!
# 注意到显然只有p^2合乎题意, 而小于10^12的p^2中, p必须小于10^6!
# 因此只需找到所有小于10^6的素数, 把他们平方并放进一个列表,
# 然后再验证那些数不在列表里即可!
# 此时复杂度~o(10^6)+o(10^6)+o(10^5)~o(10^6)~
n=10**6
primes=[]
status=[True]*(n+1)#用欧拉筛的方法找素数, 只用o(n)! (原理见py0.3)
for i in range(2,n+1):#这一步复杂度~o(10^6)
    if status[i]==True:
        primes.append(i)
        for prime in primes:
            if prime*i > n:
                break
            else:
                status[prime*i]=False
            if i % prime == 0:
                break
prime_squared=set()#注意!! 在集合中检索元素的时间复杂度为o(1), 所以要用集合!!
#但是, 要小心: 如果直接与my_set={}, 系统会识别为字典! 因此要写为=set()!!
for prime in primes:#这一步复杂度~o(10^6)
    prime_squared.add(prime**2)#集合没有append, 只有add; 且add不会造成重复~
result=[]
m=int(input())
numbers=list(map(int,input().split()))
for number in numbers:#这一步复杂度~o(10^5)
    if number in prime_squared:
        result.append('YES')
```

### \*12559: 最大最小整数 （选做）

greedy, strings, sortings, <http://cs101.openjudge.cn/practice/12559>

思路：构造一个映射 f，对每一个数 A，若其位数是 a，定义  $f(A)=A/10^{a-1}$ ；下证：将所有这些数字按照 f(a) 从大到小排序之后，组成的就是最大数！

这是因为：首先，对于两个相邻的数 A 和 B，若他们的位数分别是 a 和 b，则 AB 大于 BA 等价于  $10^b * A + B > 10^a * B + A$ ；

这又等价于  $(10^b - 1)A > (10^a - 1)B$ ，亦即  $f(A)$  大于  $f(B)$ ！

因此，在达到最大数时，f(a) 一定是单调递减的（否则可交换使其变得更大，矛盾）；同样的，在达到最小数时，f(a) 一定是单调递增的；证毕！

代码

```
```python
```

```
```
```



```

n=int(input())
nums=list(map(int,input().split()))
function_list=[]
for A in nums:
    a=len(str(A))#要先把整数转换为字符串才能判断其位数!
    f_A=A/(10**a - 1)
    function_list.append([A,f_A])
#接下来, 对 f_A 从小到大排序即可~
sorted_list=sorted(function_list,key=lambda item:item[1])
sorted_nums=[]
for i in range(n):
    sorted_nums.append(sorted_list[i][0])
#找最大数也是一模一样, 只需要把数字按函数值倒序排列即可!
reversed_sorted_nums=sorted_nums[::-1]
#不要 reversed_sorted_nums=sorted_nums.reverse()!!
#这样不仅会改变原本的列表, 而且只会返回 none!
min="".join(map(str,sorted_nums))
max="".join(map(str,reversed_sorted_nums))
print(f"{max} {min}")

```

代码运行截图 <mark> (至少包含有"Accepted") </mark>

OpenJudge

题目ID, 标题, 描述

24n2300010821

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#46581921提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

n=int(input())
nums=list(map(int,input().split()))
function_list=[]
for A in nums:
    a=len(str(A))#要先把整数转换为字符串才能判断其位数!
    f_A=A/(10**a - 1)
    function_list.append([A,f_A])
#接下来, 对 f_A 从小到大排序即可~
sorted_list=sorted(function_list,key=lambda item:item[1])
sorted_nums=[]
for i in range(n):
    sorted_nums.append(sorted_list[i][0])
#找最大数也是一模一样, 只需要把数字按函数值倒序排列即可!
reversed_sorted_nums=sorted_nums[::-1]
#不要reversed_sorted_nums=sorted_nums.reverse()!!
#这样不仅会改变原本的列表, 而且只会返回none!
min="".join(map(str,sorted_nums))
max="".join(map(str,reversed_sorted_nums))
print(f"{max} {min}")

```

基本信息

#: 46581921

题目: 12559

提交人: 24n2300010821

内存: 3800kB

时间: 22ms

语言: Python3

提交时间: 2024-10-18 20:55:05

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## ## 2. 学习总结和收获

<mark>如果作业题目简单,有否额外练习题目,比如:0J“计概 2024fall 每日选做”、CF、LeetCode、洛谷等网站题目。</mark>

这次作业前四题难度都比较简单,很快就能做出来;虽然这四道题目全都是之前做过的每日选做题,之前 AC 过一次,但是我还是都重做了一遍,来加深自己对题目的理解;

而后两题都比较有难度,我都是花了半个小时以上的时间研究才做出来的。其中 T-primes 那道题使用普通的“判断每个数的平方根在不在素数集合内”的方法尝试优化了多次都还是 TLE;受此影响,我决定大幅修改我的代码,并在修改的过程中学会了欧拉筛,并且作为一个数院的学生,我也花了不少时间研究透了它内部包含的的数论原理(如下图):

```
n=int(input())
primes=[]
status=['True']*(n+1)#用对应位置的T/F来记录每个数是否为素数
for i in range(2,n+1):
    if status[i]=='True':#如果轮到i的时候仍没被扔掉,则一定是素数!
        primes.append(i)
        #接下来进行下一步操作:把i的倍数扔掉
        # (欧拉筛的第一步简化:只扔i的素数倍数!因为此时所有的合数仍会被扔掉~)
        for prime in primes:
            if prime*i > n:
                break
            else:
                status[prime*i]='False'#把i的p倍扔掉
                # (欧拉筛的关键第二步简化:只扔i的“小于等于i的最小素因数”的素数倍数! )
                #因为这样会确保每个合数只被扔掉一次!! (原理见最后方)
                if i % prime == 0:
                    break
print(primes)
```

#下方为第二步原理:

#设k为一个合数, p为其最小素因子,  $k=tp$ ; 下证: k在、且只会在 $p*t$ 那一步被扔掉。

#一方面, 因为t的最小素因子肯定大于等于p, 所以一定存在 $p*t$ 这一步;

#另一方面, 如果k还在另外一步 $m*q$  (q为素数) 被筛去了,

#则p必整除m, 因此m最小素因子一定是p; 但q大于p! 因此矛盾! 因此只会在 $p*t$ 这一步被扔掉~

除此以外, 我也了解到了“在 TLE 的时候可以把列表转化为集合, 来减少查找一个元素在不在这个列表里的时间复杂度”的这一个技巧, 直接把循环里的  $O(n)$  变成了  $O(1)$ !

而最大最小数那道题我一开始也没有什么很好的计算机方法, 后来, 在跟我一个舍友(也是数院的)同学交流了之后, 得出了一个非常数学的一个方法(构造一个映射 f, 对每一个数 A, 若其位数是 a, 定义  $f(A)=A/10^a-1$ ; 则将这些数字按照 f(a) 从大到小排序之后, 组成的就是最大数), 感觉无论是原理还是证明都非常的漂亮~