*# Assignment #4: T-primes + 贪心*

Updated 0337 GMT+8 Oct 15, 2024

2024 fall, Complied by

**付耀贤 信息管理系 2400016634**

*## 1. 题目*

*### 34B. Sale*

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

思路

比较简单的排序和贪婪，可以接受

代码

N,M=map(int,input().split())

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

ef=[]

for \_ in price:

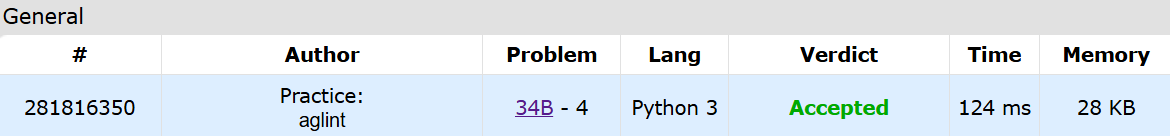
if \_<0:

ef.append(\_)

x=sum(ef[0:M])

print(-x)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### 160A. Twins*

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

思路：

比较简单，在临界条件有些小小的易错点

代码

n=int(input())

num=sorted(list(map(int,input().split())),reverse=True)

c=sum(num)

re=[]

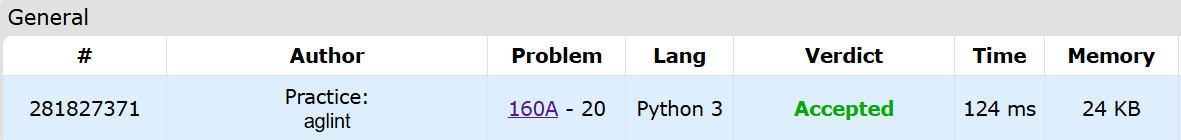
for \_ in num:

if sum(re)<=(c/2):

re.append(\_)

print(len(re))

代码运行截图 ==（至少包含有"Accepted"）==



*### 1879B. Chips on the Board*

constructive algorithms, greedy, 900,

https://codeforces.com/problemset/problem/1879/B

思路：

说真的，我觉得这道题挺难的，至少我在考场上大概率做不出来。这道题是完全在考思维，需要脑筋转得比较快。

代码

n=int(input())

for \_ in range(n):

square=int(input())

row=list(map(int,input().strip().split()))

line=list(map(int,input().strip().split()))

a=sum(row)

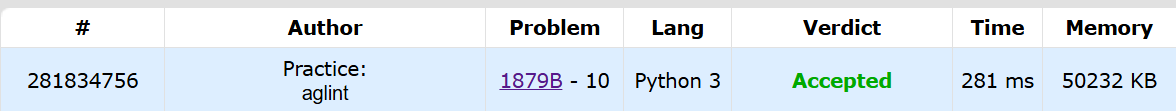
b=sum(line)

c=a+min(line)\*square

d=b+min(row)\*square

print(min(c,d))

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### 158B. Taxi*

\*special problem, greedy, implementation, 1100,

https://codeforces.com/problemset/problem/158/B

思路：

两句话：真的不难！真的写不对！

代码

import math

n=int(input())

l=sorted(list(map(int,input().split())))

count\_dict={i:0 for i in range(5)}

c=0

for num in l:

count\_dict[num]+=1

a=count\_dict[4]

if count\_dict[1]==count\_dict[3]:

a+=count\_dict[3]

elif count\_dict[1]>count\_dict[3]:

a+=count\_dict[3]

c=count\_dict[1]-count\_dict[3]

elif count\_dict[1]<count\_dict[3]:

a+=count\_dict[1]

a+=count\_dict[3]-count\_dict[1]

if count\_dict[2]%2==0:

a+=count\_dict[2]//2

a+=math.ceil(c/4)

else:

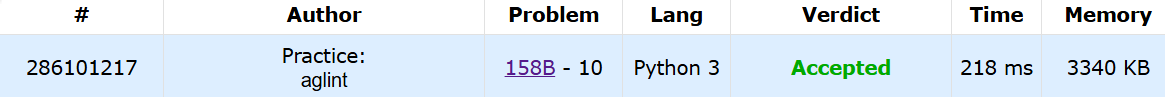
a+=count\_dict[2]//2

a+=1

if c>2:

a += math.ceil((c - 2) / 4)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### \*230B. T-primes（选做）*

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

思路：

这道题思路都会，就是如何优化。从题解挑了一个比较好理解的代码，很有收获。

代码

def euler\_sieve(n):

is\_prime = [True] \* (n + 1)

is\_prime[0] = is\_prime[1] = False

primes = []

for i in range(2, n + 1):

if is\_prime[i]:

primes.append(i)

for p in primes:

if i \* p > n:

break

is\_prime[i \* p] = False

if i % p == 0:

break

return is\_prime

s = euler\_sieve(1000000)

input()

for i in map(int, input().split()):

sqrt\_i = i \*\* 0.5

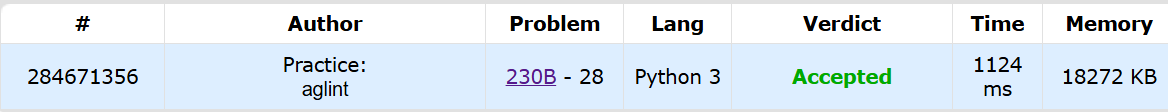
if sqrt\_i % 1 == 0 and s[int(sqrt\_i)]:

print('YES')

else:

print('NO')

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*### \*12559: 最大最小整数 （选做）*

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

思路：其实不难，关键是把位数化统一

代码

def find(n, numbers):

num= list(map(str, numbers))

max\_num = ''.join(sorted(num, key=lambda x: x \* 10, reverse=True))

min\_num = ''.join(sorted(num, key=lambda x: x \* 10))

return max\_num, min\_num

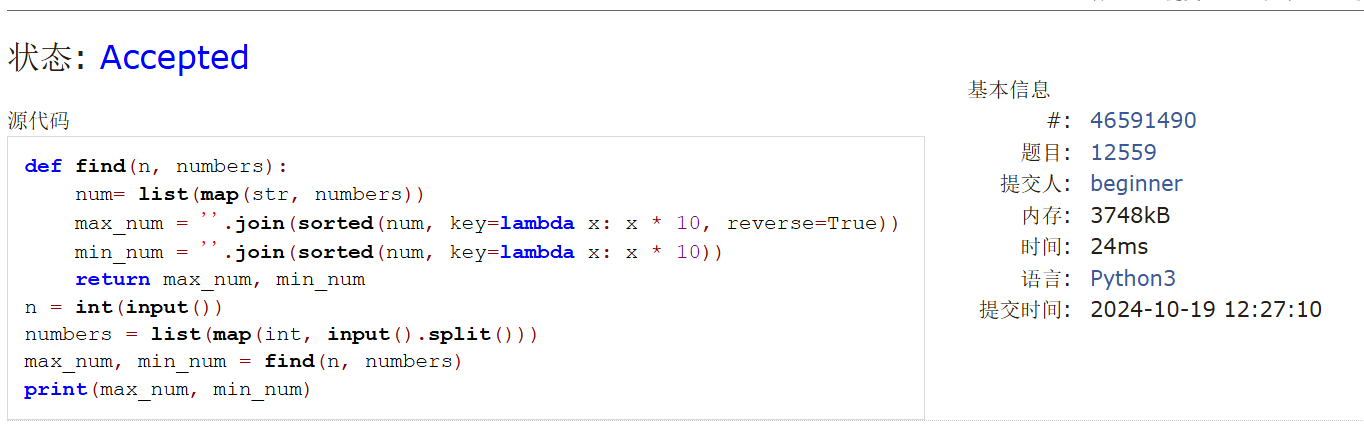
n = int(input())

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

max\_num, min\_num = find(n, numbers)

print(max\_num, min\_num)

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



*## 2. 学习总结和收获*

最近在做1100左右难度的题目，大概涉及贪心、动态规划。真的感觉难度上来了，要么是思路很难想到、要么是写不对、要么是超时，在一堆每日选做里只能AC个别题目，看题解也是比较费力。按自己的节奏来，慢慢理解吧，能做多少是多少！