# # Assignment #4: T-primes + 贪心

Updated 0337 GMT+8 Oct 15, 2024

2024 fall, Complied by 颜鼎堃 工学院

#### 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

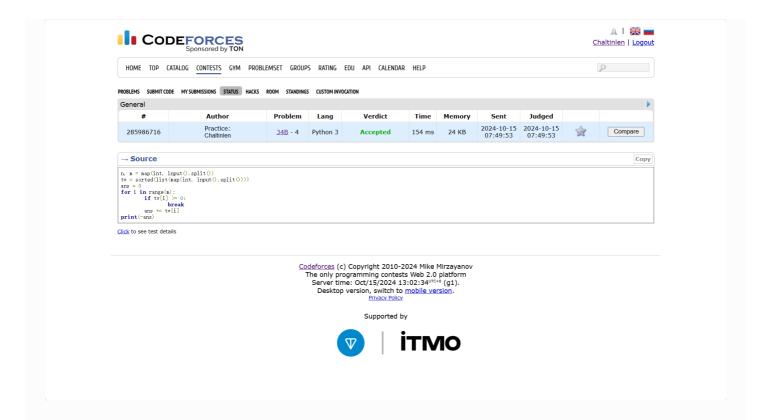
greedy, sorting, 900, https://codeforces.com/problemset/problem/34/B

#### 思路:

• 注意如果负值不到m个,取完所有负值就应当停止

#### 代码

```
n, m = map(int, input().split())
tv = sorted(list(map(int, input().split())))
ans = 0
for i in range(m):
    if tv[i] >= 0:
        break
    ans += tv[i]
print(-ans)
```



### # 160A. Twins

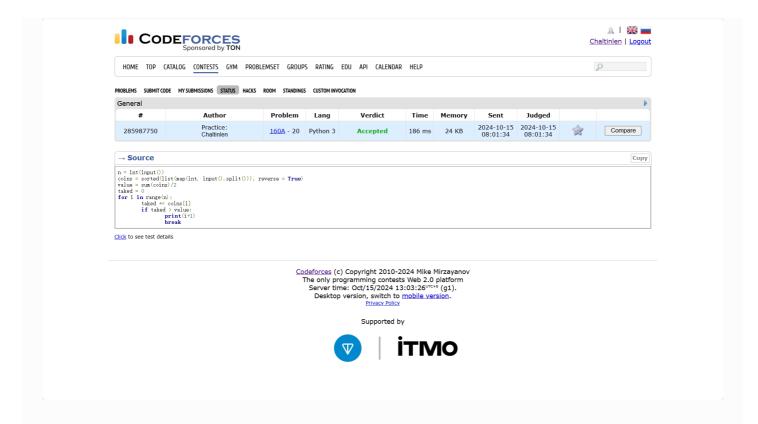
greedy, sortings, 900, https://codeforces.com/problemset/problem/160/A

#### 思路:

• 注意拿取的硬币数量是下标加1

代码

```
n = int(input())
coins = sorted(list(map(int, input().split())), reverse = True)
value = sum(coins)/2
taked = 0
for i in range(n):
   taked += coins[i]
   if taked > value:
        print(i+1)
        break
```



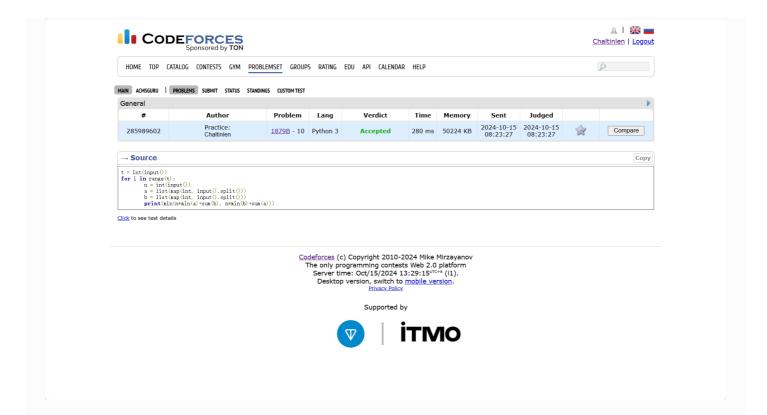
## 1879B. Chips on the Board

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

• 题目看着复杂,其实只要对着标号最小的一行(或者一列)使劲取就好了

代码

```
t = int(input())
for i in range(t):
    n = int(input())
    a = list(map(int, input().split()))
    b = list(map(int, input().split()))
    print(min(n*min(a)+sum(b), n*min(b)+sum(a)))
```



### # 158B. Taxi

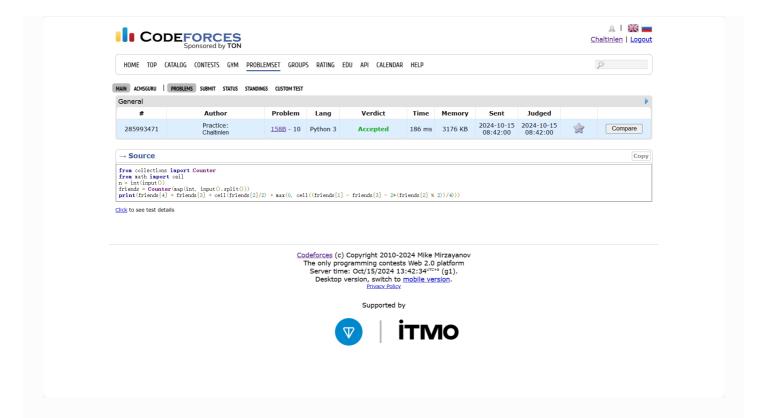
\*special problem, greedy, implementation, 1100, https://codeforces.com/problemset/problem/158/B

#### 思路:

• 装箱子简单版,想清楚怎么装就简单了

代码

```
from collections import Counter
from math import ceil
n = int(input())
friends = Counter(map(int, input().split()))
print(friends[4] + friends[3] + ceil(friends[2]/2) + max(0, ceil((friends[1] - friends[3] - 2*(friends[2] % 2))/4)))
```



# \*230B. T-primes(选做)

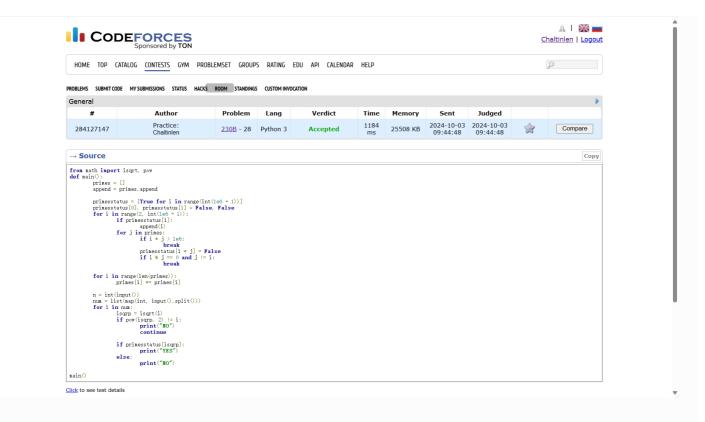
binary search, implementation, math, number theory, 1300, http://codeforces.com/problemset/problem/230/B

#### 思路:

- 欧拉筛法,在一位信息学竞赛同学的帮助下学到了
- 试除法、埃拉托色尼筛法都会超时

#### 代码

```
from math import isqrt, pow
def main():
    primes = []
    append = primes.append
    primesstatus = [True for i in range(int(1e6 + 1))]
    primesstatus[0], primesstatus[1] = False, False
    for i in range(2, int(1e6 + 1)):
        if primesstatus[i]:
            append(i)
        for j in primes:
            if i * j > 1e6:
                break
            primesstatus[i * j] = False
            if i % j == 0 and j != i:
                break
    for i in range(len(primes)):
        primes[i] *= primes[i]
    n = int(input())
    num = list(map(int, input().split()))
    for i in num:
        isqrp = isqrt(i)
        if pow(isqrp, 2) != i:
            print("NO")
            continue
        if primesstatus[isqrp]:
            print("YES")
        else:
            print("NO")
main()
```



### \*12559: 最大最小整数 (选做)

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

#### 思路:

- 经过一个下午的冥思苦想(我承认我看了测试数据来debug),最后用四行代码AC了本题
- 整体思路如下
  - 1. 对读入的数组用 groupby() 函数根据最高位数字(即1~9)进行分组,在分组前先用同样的依据进行排序,一方面是为了使得数据在分组后按从大到小的顺序(即9开头的在前,1开头的在后),另一方面也是因为 groupby() 只能将连续的同类数据归为一组
  - 2. 用生成器写法将分好的列表合到一个列表中
  - 3. 对每一个列表中的元素分别排序,在偷看了测试数据后发现不超过1000,故按照以下标准进行排列:
    - 将数据左对齐,不够的位数统一用开头数字填充,如76与765比较时,被视为767777与765777比较,显然前者更大,排在前面
    - 如果填充后大小相同,考虑第二位数字。若无第二位数字,则顺序不影响,如7和77。若有第二位数字,则考虑第二位数字和第一位数字的大小。如果第二位数字比第一位数字大,则把较短的放在前面,比如787与78;如果第二位数字比第一位数字小,则把较短的放在后面,比如989与98。
  - 4. 现在完成了对二维列表的排序,考虑进行输出。事实上,正序输出得到最大数,逆序输出得到最小数,运用好 "".join() 与 reverse() 函数即可
  - 5. 在编写代码的过程中多运用 map() 函数与 lambda 表达式

#### 6. 最后我们就得到了这四行抽象的代码

代码

```
from itertools import groupby
n = int(input())
num = list(map(lambda t: sorted(t, key = lambda s: (s.ljust(6, s[0]),
-len(s) if (len(s) > 1 and s[1] > s[0]) else len(s)), reverse = True),
[list(group) for key, group in groupby(sorted(input().split(), key =
lambda s: s[0], reverse = True), key = lambda s: s[0])]))
print("".join(map("".join, num)), "".join(list(map("".join,
reversed(list(map(reversed, num))))))))
```

### 代码运行截图 (至少包含有"Accepted")



### # 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"计概2024fall每日选做"、CF、LeetCode、洛谷等网站题目。

真心感觉自己脑子不够用,啥都得问别人,要么就要看测试数据(比如第六题,对着三千个数研究半天才意识到哪有问题),做题也做的慢,有时候WA也莫名其妙不知道为啥,我的评价是:



释义: 开始料到不及格了

