Assignment #4: T-primes + 贪心

Updated 0337 GMT+8 Oct 15, 2024

2024 fall, Complied by <mark>汤伟杰,信息管理系</mark>

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:

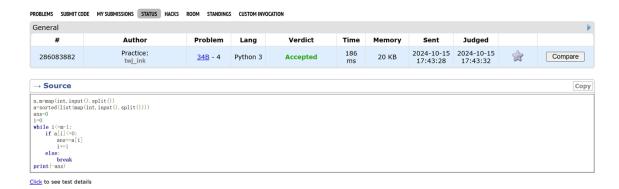
greedy 思路是先排序,再用从左到右的单指针扫描,指针用 i 的索引,套在 while 循环里。

注意题目说"至多拿走 m 个", 说明如果从小到大排序后, 前 m 个中有正数时就可以暂停了, 所以使用 while 的循环不断对 i 加 1 直到出现正数, 再把答案取绝对值即可。

代码

```
n,m=map(int,input().split())
a=sorted(list(map(int,input().split())))
ans=0
i=0
while i<=m-1:
    if a[i]<=0:
        ans+=a[i]
        i+=1
    else:
        break
print(-ans)</pre>
```

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:

greedy 思路是先逆向排序,再用从左到右单指针 i 套在 while 循环里,不断更新比较左边数字是否大于了右边数字。

代码

```
n=int(input())
a=list(map(int,input().split()))
a.sort(reverse=True)
i=0
sum_one=sum(a[:1])
sum_two=sum(a[1:])
while sum_one<=sum_two:
    i+=1
    sum_one+=a[i]
    sum_two-=a[i]
print(i+1)</pre>
```

代码运行截图 == (至少包含有"Accepted") ==



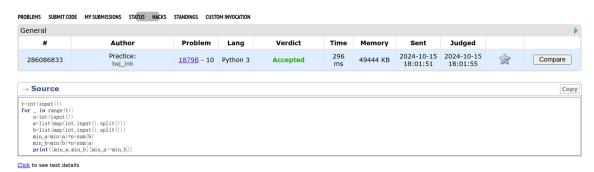
1879B. Chips on the Board

constructive algorithms, greedy, 900, https://codeforces.com/problemset/problem/1879/B 思路: greedy 思路是选定最小数字的 row 或者 col,全部放在他们身上,两个数字的较小者就是答案。

代码

```
t=int(input())
for _ in range(t):
    n=int(input())
    a=list(map(int,input().split()))
    b=list(map(int,input().split()))
    min_a=min(a)*n+sum(b)
    min_b=min(b)*n+sum(a)
    print([min_a,min_b][min_a>=min_b])
```

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



158B. Taxi

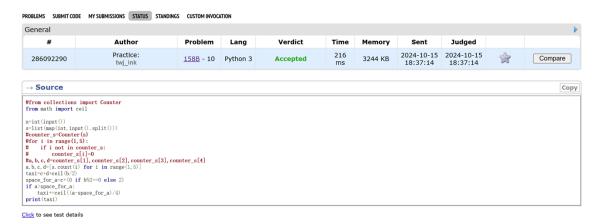
*special problem, greedy, implementation, 1100, https://codeforces.com/problemset/problem/158/B

思路:

easy 版的装箱子问题,同时看了 cf 某个红名大佬的代码对 abcd 的赋值做了简化

代码

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:

欧拉筛的思路,然后 def 一个判断是否为完全平均数的函数,return 里面除了布尔值外,还返回一个开根号之后的 root 以便判断这个 root 是否是质数。

代码

```
import math
def euler sieve(max num):
    """使用欧拉筛法生成指定范围内的质数集合"""
    is_prime = [True] * (max_num + 1)
    primes = []
   for i in range(2, max_num + 1):
       if is_prime[i]:
           primes.append(i)
       for p in primes:
           if p * i > max_num: # 超出范围停止
           is_prime[p * i] = False
           if i % p == 0: # 确保每个质数只用一次
               break
    return set(primes)
def is_perfect_square(x):
   if x < 0:
       return False
    root = math.isqrt(x)
    return root * root == x, root
n = int(input())
s = list(map(int, input().split()))
```

```
# 预先计算所有小于等于 sqrt(最大值) 的质数
max_value = max(s)
max_sqrt = int(math.isqrt(max_value))
prime_set = euler_sieve(max_sqrt)

for i in s:
    perfect_square, root = is_perfect_square(i)
    if perfect_square:
        print('YES' if root in prime_set else 'NO')
    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:

不小心看到群里的倍长了,所以会想到对每个数字先循环几遍(但是题目没说每个数字的大小范围,故随便选择了循环 5 次),然后直接采用字典序排序,但是最终答案需要未循环的字符串,所以可以先将原字符串和循环的字符串组成一个元组一起保存,按照索引 1 排序,利用索引 0 组合答案。

(代码一遍过了超级开心

代码

```
n=int(input())
s=list(map(str,input().split()))
new_s=[]
ans=[]
for i in s:
    j=i*5
    new_s.append((i,j))
new_s.sort(key=lambda x:x[1],reverse=True)
for i,j in new_s:
    ans.append(i)
print(''.join(ans),''.join(ans[::-1]))
```

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

状态: Accepted

```
基本信息
n=int(input())
s=list(map(str,input().split()))
new s=[]
ans=[]
for i in s:
   i=i*5
   new s.append((i,j))
new s.sort(key=lambda x:x[1], reverse=True)
for i, j in new s:
   ans.append(i)
print(''.join(ans),''.join(ans[::-1]))
```

#: 46512372 题目: 12559 提交人: 24n2400016635

内存: 3768kB 时间: 20ms 语言: Python3

提交时间: 2024-10-16 00:03:46

©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:OJ"计概 2024fall 每日选做"、CF、 LeetCode、洛谷等网站题目。</mark>

感觉前三道 greedy 比较简单,思路好像代码也容易实现; 第四道是装箱子的简单版本; 第五题考欧拉筛;第六题考字符串的字典序排序(ASCII表大小排序),我一个0基础做 起来居然障碍不是太大太大还是有点惊讶的。

不过每日选做的 greedy 题目,自己会先有思路并用复杂代码实现,验证后发现思路是错 误的,这时就很难再去想其他思路了;感觉这种题目对于我来说要么一遍过,要么一直过 不了,很难,很棘手,但是常常也是束手无策。等过几天课少了再集中去刷 greedy 的题 目,看看能不能找到经验,要不然现在做起来太难受了。