

Assignment #4: T-primes + 贪心

Updated 0337 GMT+8 Oct 15, 2024

2024 fall, Compiled by 胡杨 元培学院

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：要赚更多的钱，那么就要把所有能带走的负数加起来再取绝对值！

先创立所有电视机的列表，对其排序，保留小于零的元素，把前m项相加取绝对值即可

代码

```
n,m=map(int,input().split())
l=list(map(int,input().split()))
l.sort()

l_buy=[i for i in l if i<=0]
money=abs(sum(l_buy[:m]))

print(money)
```

代码运行截图 (至少包含有"Accepted")

286334875	Oct/17/2024 15:28 ^{UTC+8}	Itseventeen	34B - Sale	Python 3	Accepted	154 ms	0 KB
-----------	------------------------------------	-------------	------------	----------	----------	--------	------

160A. Twins

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

思路：建立所有硬币值的列表s，将其倒序排序，然后求和为a，看前多少项的和 $>a/2$

题解启示：题解第一个使用的方法是建立硬币价值0-100的数组，计数每个面值硬币的数量，再看多少项 $\geq a/2$ 。

助教老师指出这种方法发时间复杂度应该要小一点，是 $O(n)$ ，我的方法的复杂度是 $O(n\log n)$ ，但是能过就行XD

代码

```
n=int(input())
s=list(map(int,input().split()))
s.sort(reverse=True)
a=sum(s)

b=0 #初始化我的钱的总和
num=0 #初始化我拿到的硬币的数量
for i in range(n):
    if b>(a//2):
        break
    b+=s[i]
    num+=1

print(num)
```

代码运行截图 (至少包含有"Accepted")

286337343	Oct/17/2024 15:46 UTC+8	ltseventeen	160A - Twins	Python 3	Accepted	154 ms	0 KB
-----------	-------------------------	-------------	--------------	----------	----------	--------	------

1879B. Chips on the Board

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

思路：要满足题目要求，则至少需要放n个chip，它们位于不同的行，或者不同的列

可以找出a,b两列数中最小的数， a_{\min}, b_{\min} ，比较 $n*a_{\min}+\text{sum}(b)$ (chips位于最小行的方案) 和 $n*b_{\min}+\text{sum}(a)$ (chips位于最小列的方案) 的大小，其较小值即为答案

代码

```

t=int(input())

for _ in range(t):
    n=int(input())
    a=list(map(int,input().split()))
    b=list(map(int,input().split()))

    a_min=min(a)
    b_min=min(b)
    answer1=a_min*n+sum(b) #位于最小行的方案
    answer2=b_min*n+sum(a) #位于最小列的方案
    print(min(answer1,answer2))

```

代码运行截图 (至少包含有"Accepted")

286342545	Oct/17/2024 16:24 UTC+8	Itseventeen	1879B - Chips on the Board	Python 3	Accepted	265 ms	50700 KB
-----------	-------------------------	-------------	----------------------------	----------	----------	--------	----------

158B. Taxi

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

思路：可以设置一个数组people=[0]*5，计数有1, 2, 3, 4人的小组分别有几个，然后四人小组一组一辆车，三人组和一人组拼车，若三人组有剩余则独自坐车，若一人组有剩余再和剩下的两人组拼车。2人组和2人组拼车，最后单出来的2人组和2个一人组拼车。如果还有剩余的1人组就4个1人组拼车

题解启示：题解思路一致，但是进行了数学优化，并且通过 `a,b,c,d=map(input().count, ('1','2','3','4'))`，直接把1234人数组的数量数出来了

```
d+c+(b*2+max(0,a-c)+3)//4
```

d: 四人组, c:三人组, 比一人组多也要+c,比一人组少也要+c再讨论a-c

b*2二人组的总人数，偶数直接//4就行，奇数也就剩一组两个人，再看一人组的情况

+3保证了二人组剩两个人而一人组没人或只有一个人时，或者二人组没人一人组有1-3人时都有一辆车

代码

```

import math
n=int(input())
a=list(map(int,input().split()))

people=[0]*5 #初始化数组，第0-4个数分别代表有0-4个人的小组
for i in a:
    people[i]+=1 #统计每个小组的人数

count=0 #初始化出租车计数器

```

```

count+=people[4] #将4人小组的出租车数加到计数器上

if people[3]>=people[1]:
    count+=people[3] #将3人小组和1人小组拼车并带走，剩余的3人小组单独打一辆车
    count+=math.ceil(people[2]/2)#将2人小组拼车，每2组一辆车，向上取整

else:
    count+=people[3] #将3人小组和1人小组拼车并带走
    people[1] -= people[3] # 计算1人小组剩余的组数

    count+=people[2]//2 #将2人小组拼车，每2组一辆车

    if people[2]%2!=0: #如果2人小组人数为奇数，则最后一辆车和1人小组拼
        count+=1
        people[1]-=2
        if people[1]>0: #如果1人小组还有人
            count+=math.ceil(people[1]/4)

    else: #如果2人小组人数为偶数，则内部消耗完全
        count+=math.ceil(people[1]/4) #每4组一辆车，向上取整

print(count)

```

代码运行截图 (至少包含有"Accepted")

286345954	Oct/17/2024 16:47 UTC+8	ltseventeen	158B - Taxi	Python 3	Accepted	186 ms	3200 KB
-----------	-------------------------	-------------	-------------	----------	----------	--------	---------

*230B. T-primes (选做)

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

思路:

代码

代码运行截图 (至少包含有"Accepted")

*12559: 最大最小整数（选做）

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

思路：

代码

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。

之前其实一直以为贪心递归什么的是一种新的代码，就像学会加减乘除过后来学解方程组，现在发现好像主要是一种思考问题的方式，一种思想类型的东西。贪心就是每一步都要最优然后达到全局最优的样子？但是怎么就能够保证每一步最优就会达到全局最优呢这也是个问题。

本周也是身体爆炸的一周，伴随着各种各样期中考试接近，前面四道题解决得比较顺利，但是我贪心的讲义还没做，准备先把讲义做完再来补后面两道选做的作业）

学不完了学不完了为什么我同时有那么多硬课啊（惨叫）