

# Assignment #4: T-primes + 贪心

---

Updated 1814 GMT+8 Sep 30, 2025

2025 fall, Complied by 韩旭 元培学院

## 说明:

### 1. 解题与记录:

对于每一个题目, 请提供其解题思路 (可选), 并附上使用Python或C++编写的源代码 (确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted)。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。(推荐使用Typora <https://typoraio.cn> 进行编辑, 当然你也可以选择Word。) 无论题目是否已通过, 请标明每个题目大致花费的时间。

2. 提交安排: \*\*提交时, 请首先上传PDF格式的文件, 并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像, 提交的文件为PDF格式, 并且“作业评论”区包含上传的.md或.doc附件。
3. **延迟提交:** 如果你预计无法在截止日期前提交作业, 请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业, 以保证顺利完成课程要求。

## 1. 题目

---

### 34B. Sale

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

时间: 10min

#### 思路

首先, 把价格为负的电视挑出来, 然后取其中能拿回去的最大值即可。

注意这里面最好直接逆向排序, 需要用到sort函数的reverse选项。

#### 代码

```

n,m=map(int,input().split())
neg=[]
ans=0
L=list(map(int,input().split()))
for i in L:
    if i<0:
        neg.append(abs(i))
    else: continue
neg.sort(reverse=True)
for j in range(min(m,len(neg))):
    ans+=neg[j]
print(ans)

```

## 代码运行截图

342416094	Oct/07/2025 17:48 <sup>UTC+8</sup>	xuhanecon	34B - Sale	Python 3	Accepted	186 ms	0 KB
-----------	------------------------------------	-----------	------------	----------	----------	--------	------

## 160A. Twins

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

用时：10min

## 思路

和上一题很像，先从大到小进行排序，然后一直拿直到多于一半。注意等于一半的情况还要继续拿。

## 代码

```

n=int(input())
L=list(map(int,input().split()))
S=sum(L)
L.sort(reverse=True)
total=0
count=0
while total<=S-total:
    total+=L[count]
    count+=1
print(count)

```

## 代码运行截图

342417102	Oct/07/2025 17:56 <sup>UTC+8</sup>	xuhanecon	160A - Twins	Python 3	Accepted	186 ms	0 KB
-----------	------------------------------------	-----------	--------------	----------	----------	--------	------

## 1879B. Chips on the Board

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

用时：15min

思路

这道题想出逻辑等价形式后，代码并不困难。要想让每个cell对应的行或列上至少有一个chip，等价于chips要么每行都有，要么每列都有。所以成本最小的方案只有两种可能：（1）取每行的成本，加上n倍成本最小的列的成本；（2）取每列的成本，加上n倍成本最小的行的成本。这个题的example略有点误导性，因为这里有两个成本最小的行，所以它在占满每个列的同时没有放在同一行上。

代码

```
N=int(input())
for _ in range(N):
    n=int(input())
    row=list(map(int,input().split()))
    column=list(map(int, input().split()))
    ans=min(sum(row)+n*min(column), sum(column)+n*min(row))
    print(ans)
```

代码运行截图

342420570	Oct/07/2025 18:24 UTC+8	xuhanecon	1879B - Chips on the Board	Python 3	Accepted	264 ms	50800 KB
-----------	-------------------------	-----------	----------------------------	----------	----------	--------	----------

## M01017: 装箱问题

greedy, <http://cs101.openjudge.cn/pctbook/M01017/>

用时：25min

思路

这题是二维版的Taxi。基本思路也是贪心，即优先装大的箱子。但具体讨论很繁琐，尤其需要注意几个点：

- (1) 在装到3\*3以及更小的箱子时，涉及整除的问题。比如，每个6\*6的大箱子最多能装4个小箱子，那么对于n个3\*3的小箱子，所需要的大箱子个数应该是 $(n+3)//4$ ，而不是 $n//4$ 或者 $n//4+1$ 。
- (2) 此外，在计算剩余空间的时候，如果不存在剩余空间（余数 $r=0$ ），需要单独分类，因为其算式和 $r!=0$ 是不同的。

代码

```

while True:
    L=list(map(int,input().split()))
    if sum(L)==0:
        break
    else:
        #Put 6*6
        ans=L[5]

        # Put 5*5
        ans+=L[4]
        L[0]=max(L[0]-L[4]*11,0)

        # Put 4*4
        ans+=L[3]
        need2=5*L[3]
        if L[1]>=need2:
            L[1]-=need2
        else:
            left=(need2-L[1])*4
            L[1]=0
            L[0]=max(L[0]-left,0)

        #Put 3*3
        ans+=(L[2]+3)//4
        r=L[2]%4
        if r:
            if r==1:
                need2=5
                remain1=7
            elif r==2:
                need2=3
                remain1=6
            else:
                need2=1
                remain1=5

            if L[1]>=need2:
                L[1]-=need2
                L[0]=max(L[0] - remain1, 0)
            else:
                remain1+=(need2 - L[1])*4
                L[1] = 0
                L[0] = max(L[0] - remain1, 0)

        #Put 2*2
        ans+=(L[1]+8)//9
        r=L[1]%9
        if r:
            left_area=36-r*4
            L[0] = max(L[0] - left_area, 0)

        #Put 1*1
        ans+=(L[0]+35)//36

```

```
print(ans)
```

## 代码运行截图

### #50248995提交状态

查看 提交 统计 提问

状态: Accepted

#### 源代码

```
while True:
    L=list(map(int,input().split()))
    if sum(L)==0:
        break
    else:
        #Put 6*6
        ans=L[5]

        # Put 5*5
        ans+=L[4]
        L[0]=max(L[0]-L[4]*11,0)

        # Put 4*4
        ans+=L[3]
        need2=5*L[3]
        if L[1]>=need2:
            L[1]-=need2
        else:
            left=(need2-L[1])*4
            L[1]=0
            L[0]=max(L[0]-left,0)
```

#### 基本信息

#: 50248995  
题目: M01017  
提交人: hanxu  
内存: 3748kB  
时间: 36ms  
语言: Python3  
提交时间: 2025-10-07 19:01:46

## M01008: Maya Calendar

implementation, <http://cs101.openjudge.cn/practice/01008/>

用时: 40min

## 思路

理解题目后思路并不困难，主要是计算出总天数之后再取余数。但是算余数的时候余数为0的情况均需要特别讨论。本题情景生僻，所以debug比较困难，用时较久。

## 代码

```
haab_month=[  
    "pop", "no", "zip", "zotz", "tzec", "xul",  
    "yoxkin", "mol", "chen", "yax", "zac", "ceh",  
    "mac", "kankin", "muan", "pax", "koyab", "cumhu", "uayet"  
]  
haab_dict=dict(zip(haab_month, list(range(0,19))))  
  
tz_names=[
```

```
"imix", "ik", "akbal", "kan", "chicchan",
"cimi", "manik", "lamat", "muluk", "ok",
"chuen", "eb", "ben", "ix", "mem",
"cib", "caban", "eznab", "canac", "ahau"
]
tz_dict=dict(zip(list(range(1,21)),tz_names))

N=int(input())
print(N)
for _ in range(N):
    day, month, year=input().split()
    day=int(day[:-1])
    month=int(haab_dict[month])
    year=int(year)
    total=year*365+month*20+(day+1)
    tz_year=total//260
    remain=total%260
    if remain==0:
        tz_year-=1
        tz_num=13
        tz_name=20
    else:
        tz_num=remain%13
        if tz_num==0:
            tz_num=13
        tz_name=remain%20
        if tz_name==0:
            tz_name=20
    tz_name=tz_names[tz_name-1]
    print(tz_num,tz_name,tz_year)
```

代码运行截图

状态: Accepted

源代码

```

haab_month=[  
    "pop", "no", "zip", "zotz", "tzec", "xul",  
    "yoxkin", "mol", "chen", "yax", "zac", "ceh",  
    "mac", "kankin", "muan", "pax", "koyab", "cumhu", "uayet"  
]  
haab_dict=dict(zip(haab_month, list(range(0,19))))  
  
tz_names=[  
    "imix", "ik", "akbal", "kan", "chicchan",  
    "cimi", "manik", "lamat", "muluk", "ok",  
    "chuen", "eb", "ben", "ix", "mem",  
    "cib", "caban", "eznab", "canac", "ahau"  
]  
tz_dict=dict(zip(list(range(1,21)), tz_names))  
  
N=int(input())  
print(N)  
for _ in range(N):  
    day, month, year=input().split()  
    day=int(day[:-1])  
    month=int(haab_dict[month])  
    year=int(year)  
    total=year*365+month*20+(day+1)  
    tz_year=total//260  
    remain=total%260  
    if remain==0:  
        tz_year-=1  
        tz_num=13  
        tz_name=20  
    else:  
        tz_num=remain%13  
        if tz_num==0:  
            tz_num=13  
        tz_name=remain%20  
        if tz_name==0:  
            tz_name=20  
    tz_name=tz_names[tz_name-1]  
    print(tz_num, tz_name, tz_year)

```

## 基本信息

#: 50251047  
 题目: 01008  
 提交人: hanxu  
 内存: 4044kB  
 时间: 25ms  
 语言: Python3  
 提交时间: 2025-10-07 21:50:46

## 230B. T-primes (选做)

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

本题暂时跳过。

## 2. 学习总结和收获

- 贪心的共同点:** 先排序，再做“就近最优”。比如第一题Sale 先挑负价、按绝对值降序拿；第二题Twins 按硬币降序累加到严格“大于一半”为止。写法相对比较直接，关键是一些细节。
- 配对的思路更省分支:** 像装箱这类（以及之前的Taxi），把能凑满的优先配，剩下再用最小单位补。AI说装箱问题如果用固定“余数—可补空间”的表，更不易错。

- **两种方案取较小**: 第三题的思路转化很有意思，把逻辑约束转化为少数几种可能性，最后做比较。
- **边界细节**: Maya Calendar 主要错在“从 0 开始、余数为 0 要不要特判”上。AI 建议可以改成“总天数 0 基 + 取模直取”更加方便，最多可能需要把结果+1，这可能也是编程语言从 0 开始计数的好处。
- **T-primes**看上去思路不太难，就是平方根是质数的完全平方数，但是可能需要用筛法来加速质数的筛选，所以本质上还是一个筛质数的问题。后面几周可能会补上。