

DP from USACO

Wizard

1. 单调队列优化

- ① 土地并购 (Land Acquisition, 2008 Mar)
- ② 干草塔 (Tower of Hay, 2009 Open)
- ③ 又买饲料 (Buying Feed, 2010 Nov)
- ④ 玉米实验 (Cornfields, 2003 Mar)
- ⑤ 修剪草坪 (Mowing the Lawn, 2011 Open)

2. 树型

- ① 焊接 (Soldering, 2011 Open)
- ② 产奶比赛 (Milk Team Select, 2006 Mar)
- ③ 道路重建 (Rebuilding Roads, Feb 2002)
- ④ 手机网络 (Cell Phone Network, 2008 Jan)

3. 背包问题续

- ① 电子游戏 (Video Game Troubles, 2009 Dec)
- ② 最少找零 (The Fewest Coins, 2006 Dec)
- ③ 三个代表 (Jersey Politics, 2005 Feb)
- ④ 录制唱片 (Raucous Rockers, 1996 Qualifying Round)

4. 背包问题

- ① 股票市场 (Stock Market, 2009 Feb)
- ② 奶牛会展 (Cow Exhibition, 2003 Fall)
- ③ 太空电梯 (Space Elevator, 2005 Mar)

④ 平分子集 (Subset Sums, 1998 Spring)

5. 区间型

① 提交作业 (Turning in Homework, 2004 Open)

② 抢鲜草 (Grazing on the Run, 2005 Nov)

③ 最优回文 (Cheapest Palindrome, 2007 Open)

④ 智取金币 (Treasure Chest, 2010 Dec)

6. 其他一

① 打扫食槽 (Cleaning Up, 2009 Mar)

② 奶牛自行车队 (Cow Cycling, Feb 2002)

③ 滑雪缆车 (Ski Lift, 2006 Mar)

④ 奶牛飞盘队 (Cow Frisbee Team, 2009 Mar)

7. 其他二

① 滑雪比赛 (Bobsledding, 2009 Dec)

② 滑雪课程 (Ski Lessons, 2009 Open)

③ 方形牛棚 (Big Barn, 1997 Fall)

④ 接住苹果 (Apple Catching, 2004 Nov)

⑤ 公司利润 (Profits, 2011 Jan)

土地并购

(Land Acquisition, 2008 Mar)

约翰准备扩大他的农场，眼下必须购买 N 块长方形的土地。如果约翰单买一块土地，价格就是土地的面积。他也可以选择并购一组土地，并购的价格为这些土地中最大的长乘以最大的宽。比如约翰并购一块 3×5 和一块 5×3 的土地，他只需要支付 $5 \times 5 = 25$ 元，比单买合算。约翰发现，将这些土地分成不同的小组来并购可以节省经费。给定每份土地的尺寸，请你帮助他计算购买所有土地所需的最小费用。

输入格式

- 第一行：一个整数： N ， $1 \leq N \leq 50000$
- 第二行到第 $M+1$ 行：第 $i+1$ 行有两个整数，代表第 i 块土地的长度 H_i 和宽度 W_i ， $1 \leq H_i, W_i \leq 10^6$

输出格式

- 单个整数：购买所有土地的最小费用

首先我们按长与宽都递减の排序，如果有一个矩形长宽都不如另一个矩形，那么可以忽略它。

剩下的矩形可以看做，长度递增而宽度递减。

$F[i]$ 表示前 i 个矩形的最小花费。

那么有 $f[i] = \min(f[j] + x[j+1] * y[i]) (j < i)$

当然 $O(N^2)$ 的算法是对不完的。

斜率优化不再赘述。

干草塔

(Tower of Hay, 2009 Open)

显然，将塔看作一个面积一定图形，要使其最高，必须最瘦。

$F[i]$ 表示从 $n \rightarrow i$ 层的最底层的宽度。

$F[i] = \min(\text{sum}[j-1]) - \text{sum}[i-1]; (N > j > i)$

显然， j 越小 $F[i]$ 越优。

$G[i]$ 自然用来记 $n \rightarrow i$ 最多的层数。

当然 $F[i] \geq F[j]$ ，即： $F[j] \leq \min(\text{sum}[j-1]) - \text{sum}[i-1]$ 。

$\text{Sum}[i-1] \leq \text{sum}[j-1] - f[j]$ ， $\text{sum}[i-1]$ 是递减的，所以当第一个 i 满足条件后，之后的 $i-1 \sim 1$ 都是满足的。

```
for (int i=n;i--i) {  
    while (h<t && f[q[h+1]]<=sum[q[h+1]-1]-sum[i-1]) h++;  
    //找出最后一个满足的  
    f[i]=sum[q[h]-1]-sum[i-1];g[i]=g[q[h]]+1;q[++t]=i;//入队  
    while((t>h)&&(f[q[t-1]]-sum[q[t-1]-1]+sum[q[t]-1]>f[q[t]]))  
        --t,q[t]=q[t+1];  
}
```

又买饲料

(Buying Feed, 2010 Nov)

约翰开车回家，又准备顺路买点饲料了（咦？为啥要说“又”字？）。回家的路程一共有E公里，一路上会经过K家商店，第i家店里有 F_i 吨饲料，售价为每吨 C_i 元。约翰打算一共买N吨饲料，可以保证所有商店的库存总和不会少于N。

除了购买饲料要钱，运送饲料要花油钱，约翰的卡车上如果装着X吨饲料，那么他行驶一公里会花掉 X^2 元，行驶D公里需要 DX^2 元。已知第i家店距离起点有 X_i 公里，那么约翰在哪些商店买饲料运回家，才能做到最省钱呢？

输入格式

- 第一行：三个整数：N，E和K， $1 \leq N \leq 10000$ ， $1 \leq E \leq 500$ ， $1 \leq K \leq 500$
- 第二行到N+1行：第i+1行有三个整数： X_i ， F_i 和 C_i ， $0 < X_i < E$ ， $1 \leq F_i \leq 10000$ ， $1 \leq C_i \leq 10^7$

$F[i][j]$ 表示到第i个商店时，一共买了j那么多饲料的最小花费。

$$F[i][j] = \min \{ f[i-1][k] + (j-k) * v[i-1] \} + j * j;$$
$$(k < j \ \&\& \ (j-k \leq c[i-1]))$$

用单调队列优化：

$F[i][j] = \min \{ f[i-1][k] - k * v[i-1] \} + j * v[i-1] + j * j$ 这样取最小值的循环里就只与k有关，此时维持一个 $f[i-1][k] - k * v[i-1]$ 的递减单调序列，每次选择将为 $O(1)$ 。

玉米实验

(Cornfields, 2003 Mar)

约翰决定培育新的玉米品种以提高奶牛产奶效率。他要考察一下玉米的质量差距。约翰有一块被分成 $N \times N$ 个方格的玉米地。其中第 r 行第 c 列的玉米质量为 A_{rc} 。约翰的实验只需要一块 $B \times B$ 个方格的土地。他打算寻找一块质量比较均匀的土地开始自己的实验。经过前期考察，他已经锁定了 K 个位置作为实验基地的候选，其中第 i 个位置从 R_i 行 C_i 列开始，到 $R_i + B - 1$ 行 $C_i + B - 1$ 列结束。请帮助约翰计算一下，这些候选位置上质量最高的玉米与质量最低玉米之间的差值分别是多少。

输入格式

- 第一行：三个整数 N ， B 和 K ， $1 \leq N \leq 250$ ， $1 \leq B \leq N$ ， $1 \leq K \leq 10^5$
- 第二行到 $N + 1$ 行：第 $r + 1$ 行有 N 个整数，分别表示 $A_{r,1}$ 到 $A_{r,N}$ ，每个 $A_{r,c}$ 满足 $0 \leq A_{r,c} \leq 250$
- 第 $N + 2$ 行到第 $N + K + 1$ 行：第 $N + i + 1$ 行有两个整数 R_i 和 C_i ， $1 \leq R_i \leq N$ ， $1 \leq C_i \leq N$

输出格式

- 第一行到第 K 行：第 i 行表示第 i 片候选土地中质量最高的玉米与质量最低的玉米之差

首先,我们可以使用 **RMQ** 来查询每排的最小值最大值,即初始化需要 N 个 **RMQ**, $O(N^2 \cdot \log(N))$ 是可以接受的。之后的 K 次询问,每次询问 B 排的最值, $O(N \cdot K)$ 可以 **AC**。

修剪草坪

(Mowing the Lawn, 2011 Open)

约翰让他的奶牛来修建草坪。他有 N 头奶牛，编号为1到 N 。每头奶牛的工作能力是不同的，第 i 头奶牛的能力值为 E_i 。编号连续的奶牛很熟悉，如果安排编号连续的 $K+1$ 头奶牛一起工作，她们就会密谋罢工。因此，约翰想让你帮忙挑出一组奶牛，使它们的能力值之和最大，而且不会罢工。

输入格式

- 第一行：两个整数 N 和 K ， $1 \leq N \leq 10^5$ ， $1 \leq K \leq N$
- 第二行到 $N+1$ 行：第 $i+1$ 行有一个整数，表示第 i 头牛的能力 E_i ， $1 \leq E_i \leq 10^9$

输出格式

- 单个整数，表示不会罢工的组合中最大的能力之和

$f[i]$ 表示 i 不要的最小舍弃值。

```
for(int i=0;i<=n+1;i++)
```

```
{
```

```
    if(q[s].n<i-k-1&& s<t)s++; //队列长超过 k 头指针+
```

```
    f[i]=q[s].v+a[i]; //队列开头的是最小的,因此舍弃它
```

```
    while(q[t].v>f[i]&&t>=s)t--; //维持一个递增序列
```

```
    q[++t]=Cut{f[i],i}; //入队
```

```
}
```

```
Cout<<sum-f[n+1];
```


焊接

(Soldering, 2011 Open)

奶牛打算用电线焊接出一个特殊图形，这个图形是连通的，由 N 个点， $N - 1$ 条边组成，每条边的长度都是1。焊接所用的电线要从当地的商店里买。越长的电线越贵，一条长度为 L 的电线售价为 L^2 。

奶牛们已经学会了基本的焊接方法，她们会把某条电线的的一个端点焊接到另一条电线的中间某个位置，但她们没有学会如何把两条电线的端点直接焊接起来，也没有学会怎么把电线剪断。

告诉你奶牛准备焊接的图形，请告诉奶牛怎么焊接才能最节约材料费用。

输入格式

- 第一行：单个整数 N ， $1 \leq N \leq 50000$
- 第二行到 N 行：每行有两个用空格分开的整数 A 和 B ，表示图形中的一条边， $1 \leq A \leq N$ ， $1 \leq B \leq N$

输出格式

- 第一行：单个整数，表示最少的花费，保证这个数字小于 2^{63}

产奶比赛

(Milk Team Select, 2006 Mar)

约翰的 N 头奶牛报名去参加世界产奶大奖赛(Multistate Milking Match-up, 简称 MMM)。这是一项组队赛, 队伍的人数任意, 约翰的第 i 头奶牛可以为集体贡献 C_i 的牛奶。注意有些奶牛会帮倒——弄翻其他奶牛的牛奶桶, 所以 C_i 可能是负的。所有参加比赛的奶牛毫无疑问都是女士, 记第 i 头奶牛的母亲为 P_i , 如果她的母亲没有报名, P_i 就等于零。

约翰很清楚对手的实力, 确信只要自己队伍产出的牛奶不少于 X , 就一定能够赢得比赛。MMM 举办的目的之一, 是通过竞赛中的合作来增进奶牛家庭成员之间的默契。奶牛们觉得既然自己稳操胜券, 为了表示对大赛精神的支持, 她们希望派出的队伍里多出现一些母女组合。请挑选出一只能赢得比赛的队伍, 使得其中奶牛的母女关系数目最多。如果一头奶牛和她的母亲、外祖母组成一只队伍, 这只队伍算作有两对母女关系。

输入格式

- 第一行: 两个用空格分开的整数: N 和 X , $1 \leq N \leq 500$, $1 \leq X \leq 10^7$
- 第二行到 $N + 1$ 行: 第 $i + 1$ 行有两个用空格分开的整数, 表示 C_i 和 P_i 。 $-10000 \leq C_i \leq 10000$, $0 \leq P_i \leq N$, 保证输入数据中的母女关系不会出现循环

输出格式

- 单个整数: 表示母女关系的最大数, 如果无法选出一支优胜队伍, 输出 -1

类似于道路重建, 即 $f[x][i]$ 表示节点 x 分配 i 个关系能够取得的最大奶量, 由于可以是散点, 所以可以将这 i 个关系随意分配给儿子们, 最后取最好的方案。

关于如何去最佳方案, 我们可以用背包分配。

道路重建

(Rebuilding Roads, Feb 2002)

在一场严重的地震之后，奶牛们重建了约翰的牧场。牧场里有 N 个牛棚，编号为1到 N 。这些牛棚之间有道路相连，由于奶牛们的时间不够重建更多的道路，所以目前任意两个牛棚之间有且只有一条路径。

约翰打算把奶牛安置在相邻的 P 个牛棚里，位置还没选定。他想知道下一场地震将造成多大的损失。以最坏打算来看，地震至少摧毁几条道路，才会让某处的 P 个牛棚和其他牛棚完全隔绝呢？

输入格式

- 第一行：两个正整数： N 和 P ， $1 \leq P \leq N \leq 150$
- 第二行到第 N 行：共 $N - 1$ 行，每行有两个整数，表示这两个牛棚之间存在一条道路

输出格式

- 第一行：单独一个整数，表示地震需要破坏最少道路条数

$F[i][j]$ 表示以第 i 个节点的子树要变为只有 j 个儿孙（自己也算）的树所需要剪得最少的边。

然后对儿子们进行背包分配，得出他们各留多少子孙，使花费最小。

```
for(int j=p;j>=1;j--)  
    for(int l=1;j-l>=1;l++)  
        f[x][j]=min(f[x][j],f[x][j-l]+f[to][l]-2);
```

手机网络

(Cell Phone Network, 2008 Jan)

相当于皇宫看守。

$F[x][0]$ 表示不在 x 上放但监控 x 的节点们（包括 x 自己）的最小费用；

$F[x][1]$ 表示不在 x 上放但监控 x 的节点们（不一定包括 x 自己）的最小费用；

$F[x][2]$ 表示在 x 上放且监控 x 的节点们的最小费用；

则有：

$$f(T, 0) = \min_{1 \leq k \leq n} \left\{ \sum_{\substack{i=1, \\ i \neq k}}^n \min\{f(T_i, 0), f(T_i, 2)\} + f(T_k, 2) \right\}$$

~~~~~ $T$  必须由他的儿子监控

$$f(T, 1) = \sum_{i=1}^n \min\{f(T_i, 0), f(T_i, 2)\}$$

~~~~~ $T$  不一定由他的儿子监控

$$f(T, 2) = \sum_{i=1}^n \min\{f(T_i, 0), f(T_i, 1), f(T_i, 2)\} + \omega(T)$$

~~~~~ $T$  自己监控

# 电子游戏

## (Video Game Troubles, 2009 Dec)

约翰的奶牛玩游戏成瘾！本来约翰是想把她们拖去电击治疗的，但是他发现奶牛们在玩游戏后生产了更多的牛奶，也就支持它们的做法了。

但是，奶牛在选择游戏平台上的分歧很大：有些奶牛想买 Xbox 360 来跑《光晕 3》；有的奶牛想要任天堂 Wii 来跑《明星大乱斗 X》；还有奶牛想要在 PlayStation 3 上玩《潜龙谍影 4》。约翰只有  $V$  元钱，看来不够多，要做一些取舍才行。

已知市面上有  $K$  种游戏平台，如果想玩第  $i$  个平台的游戏，必须先买一台该平台的游戏机，价格为  $C_i$ 。第  $i$  个平台上有  $S_i$  种游戏，其中第  $j$  个游戏的价格为  $P_{ij}$ ，奶牛玩过这个游戏后的产出为  $E_{ij}$ 。如果想玩同一平台上的多个游戏，只要买一台游戏机就够了。

请帮他选择买哪些游戏机和游戏，才能使奶牛产奶的效益最大？注意同一个游戏买两次是会产生双倍效益产生的。

### 输入格式

- 第一行：两个整数： $K$  和  $V$ ， $1 \leq K \leq 50$ ， $1 \leq V \leq 10^6$
- 布里斯班第二行到第  $K+1$  行：第  $i+1$  行首先有两个整数  $C_i$  和  $S_i$ ， $1 \leq C_i \leq 1000$ ， $1 \leq S_i \leq 10$ ，其次有  $G_i$  对整数  $P_{ij}$  和  $E_{ij}$ ， $1 \leq P_{ij} \leq 100$ ， $1 \leq E_{ij} \leq 10^6$

### 输出格式

- 单个整数：表示通过玩游戏可以得到的最大产出之和

即 “金明的预算方案” 加强版，每个游戏平台可带 10 个游戏，所以它们可以处理为  $2^{10}$  个不同的商品且分为一组。那么可以分为  $k$  组，每组有 1024 个游戏，每组选 1 个或 0 个。

之后即为分组背包。

# 最少找零

(The Fewest Coins, 2006 Dec)

约翰在镇上买了T元钱的东西，正在研究如何付钱。假设有N种钞票，第i种钞票的面值为 $V_i$ ，约翰身上带着这样的钞票 $C_i$ 张。商店老板罗伯是个土豪，所有种类的钞票都有无限张。他们有洁癖，所以希望在交易的时候，交换的钞票张数尽可能地少。请找出一种方法，让约翰恰好付T元钱，而且在过程中交换的货币数量最少。

## 输入格式

- 第一行：两个整数：N和T， $1 \leq N \leq 100$ ， $1 \leq T \leq 10000$
- 第二行：N个整数： $V_1$ 到 $V_N$ ， $1 \leq V_i \leq 120$
- 第三行：N个整数： $C_1$ 到 $C_N$ ， $0 \leq C_i \leq 10000$

## 输出格式

- 单个整数：表示付钱找零过程中交换的最少货币数量，如果约翰的钱不够付账，或老板没法找零，输出-1

直接将土豪的补钱视作面值为'-'的纸币，并且有无限张。那么可以当做“有限硬币问题”解决，即取得面值 $x$ 所需要的最少硬币数。



# 三个代表

(Jersey Politics, 2005 Feb)

在最近的威斯康辛州人民代表大会上，奶牛们分到了三个代表名额。威州共有 $3 \times N$ 个城市，按照选举流程，应将这些城市划分成三个选区，每个选区通过投票选出一个代表。至于哪些城市归属哪个选区，是由选举委员会决定的，唯一的要求是每个选区应恰好由 $N$ 个城市组成。

而然，这里存在一个阴谋。威斯康辛州有两种奶牛——杰尔西牛和荷斯坦牛，彼此明争暗斗，貌合神离。每个城市里，恰好有1000头奶牛参加投票，其中杰尔西牛有 $J_i$ 头，剩下的都是荷斯坦牛。杰尔西牛控制了选举委员会，它们正在密谋如何保证自己种群的代表胜出。它们的底线是至少让三个代表中的两个来自杰尔西，这就需要让至少两个选区的杰尔西牛多于荷斯坦牛。请你帮助它们设计一个划分选区的方法，来完成它们的阴谋。保证这种做法是一定存在的，如果有多种方案，输出任意一种即可。

## 输入格式

- 第一行：单个整数： $N$ ， $1 \leq N \leq 60$
- 第二行到 $3N + 1$ 行：第 $i + 1$ 行有一个整数： $J_i$ ， $0 \leq J_i \leq 1000$

## 输出格式

- 第一行到第 $N$ 行：每行一个整数，表示分在第一个选区内的城市编号
- 第 $N + 1$ 行到第 $2N$ 行：每行一个整数，表示分在第二个选区内的城市编号
- 第 $2N + 1$ 行到第 $3N$ 行：每行一个整数，表示分在第三个选区内的城市编号

类似于三角形牧场，我们只关心两个选区的 Jersey 牛是否能够超过半数。使用 01 背包拓展  $f[i][j]$  成立的情况，最后检查是否有  $i, j$  都超过 500 的成立。不过貌似方案要另找，且数据大了要爆。

# 录制唱片

(RaucousRockers, 1996QualifyingRound)

贝西得到了“破锣”摇滚乐队的 $N$ 首歌。这些歌已按照创作时间排序，其中第 $i$ 首的长度为 $T_i$ 分钟。贝西打算从中精选一些歌曲，发行 $M$ 张系列专辑。每张专辑的容量有限，最多可容纳 $C$ 分钟的音乐，而且一首歌不能分拆到两张专辑中。贝西想尽量让选入专辑的歌曲多一点。假设这些歌曲必须按照创作的时间顺序在专辑中出现，请问贝西最多能选择多少歌曲？

## 输入格式

- 第一行：三个整数： $N$ ， $C$ 和 $M$ ， $1 \leq N \leq 20$ ， $1 \leq C \leq 20$ ， $1 \leq M \leq 20$
- 第二行到第 $N + 1$ 行：第 $i + 1$ 行有一个整数 $T_i$ ， $1 \leq T_i \leq 20$

## 输出格式

- 第一行：表示可以装进专辑的最多歌曲数目

$f[i][j][k]$ 表示在前 $i$ 张唱片、录到第 $j$ 分钟、录到第 $k$ 首歌所录得最多歌曲数。

$f[i][j][k] = \max\{\text{前一分钟的歌曲数}, \text{前一首歌的歌曲数}, \text{把第 } k \text{ 首歌在当前位置放进去的歌曲数（如果可以）}\}$ 。

```
for(i=1;i<=m;i++) //第 i 张唱片
    for(j=1;j<=t;j++) //第 j 分钟
        for(k=1;k<=n;k++){ //第 k 首歌
            if(j==1) //换唱片
                f[i][j][k]=max(f[i-1][t][k], f[i][j][k-1]);
                //前一首歌的歌曲数 //前一分钟的歌曲数（下同）
            else //没有换唱片
                f[i][j][k]=max(f[i][j][k-1], f[i][j-1][k]);
                if(len[k]<j) //没有换唱片
                    f[i][j][k]=max(f[i][j][k], f[i][j-len[k]][k-1]+1);
                    //把第 k 首歌在当前位置放进去的歌曲数(下同)
                if(len[k]==j) //换唱片
                    f[i][j][k]=max(f[i][j][k], f[i-1][t][k-1]+1); }
```



# 股票市场

(Stock Market, 2009 Feb)

尽管奶牛天生谨慎，它们仍然在住房抵押信贷市场中大受打击，现在它们准备在股市上碰碰运气。贝西有内部消息，她知道 $S$ 只股票在今后 $D$ 天内的价格。

假设在一开始，她筹集了 $M$ 元钱，那么她该怎样操作才能赚到最多的钱呢？贝西在每天可以买卖多只股票，也可以多次买卖同一只股票，交易单位必须是整数，数量不限。

举一个牛市的例子。假设贝西有10元本金，股票价格如下：

| 股票 | 今天的价格 | 明天的价格 | 后天的价格 |
|----|-------|-------|-------|
| A  | 10    | 15    | 15    |
| B  | 13    | 11    | 20    |

最赚钱的做法是：今天买入 A 股 1 张，到明天把它卖掉并且买入 B 股 1 张，在后天卖掉 B 股，这样贝西就有24元了。

## 输入格式

- 第一行：三个整数： $S$ ,  $D$ 和 $M$ ,  $2 \leq S \leq 50$ ,  $2 \leq D \leq 10$ ,  $1 \leq M \leq 200000$
- 第二行到第 $S + 1$ 行：第 $i + 1$ 行有 $D$ 个整数： $P_{i,1}$ 到 $P_{i,D}$ ，表示第 $i$ 种股票在第1天到第 $D$ 天的售价， $1 \leq P_{i,j} \leq 1000$

## 输出格式

- 单个整数：表示奶牛可以获得的最大钱数，保证答案不超过500000

显然  $f[i]$  表示第  $i$  天所能拥有的最大钱数。

我们第  $i$  天赚得越多， $f[i+1]$  就会越大。但是我们不能用所有钱去买某只赚得最多的股票，因为那样可能剩下一些零钱，或是直接买次一等的但更多的反而更好。

那怎么分配呢？

将  $v[i+1]-v[i]$  视做价值，我们发现每天做一次 01 背包即可。

# 奶牛会展

(Cow Exhibition, 2003 Fall)

奶牛想证明它们是聪明而风趣的。为此，贝西筹备了一个奶牛博览会，她已经对 $N$ 头奶牛进行了面试，确定了每头奶牛的智商和情商。

贝西有权选择让哪些奶牛参加展览。由于负的智商或情商会造成负面效果，所以贝西不希望出展奶牛的智商之和小于零，或情商之和小于零。满足这两个条件下，她希望出展奶牛的智商与情商之和越大越好，请帮助贝西求出这个最大值。

## 输入格式

- 第一行：单个整数 $N$ ，表示奶牛的数量， $1 \leq N \leq 100$
- 第二行到第 $N + 1$ 行：第 $i + 1$ 行有两个整数 $S_i$ 和 $F_i$ ，表示第 $i$ 头奶牛的智商和情商， $-1000 \leq S_i \leq 1000$ ， $-1000 \leq F_i \leq 1000$

## 输出格式

- 单个整数：表示情商与智商和的最大值。贝西可以不让任何奶牛参加展览，如果这样，输出0

二维 01 背包，因为背包只能解决正数的问题，所以我们整体偏移 1000, 那么以 1000 为 0 点, 在  $f[1000][1000]$  之后寻找答案，使  $i+j$  最大即可。

# 太空电梯

(Space Elevator, 2005 Mar)

奶牛们准备升空了！为了进入轨道，他们计划建造一座太空电梯。电梯需要由各种材料堆叠而成，每块材料可以自由地叠在其他材料上面，电梯的总高度就是材料的长度之和。奶牛拥有 $N$ 种材料，第 $i$ 种材料的长度为 $H_i$ ，安全海拔为 $A_i$ ，这种规格的材料有 $C_i$ 块。所谓安全海拔，就是指第 $i$ 种型号的材料只能安放在海拔小于或等于 $A_i$ 的空间里，这是因为宇宙射线可能危及材料的安全。

假设奶牛从海拔为零的地面上开始建造太空电梯，请帮助它们用这些材料堆出尽可能高的太空电梯来吧。

## 输入格式

- 第一行：单个整数： $N$ ， $1 \leq N \leq 400$
- 第二行到第 $K + 1$ 行：每行三个整数： $H_i$ ， $A_i$ ， $C_i$ ， $1 \leq H_i \leq 100$ ， $1 \leq A_i \leq 40000$ ， $1 \leq C_i \leq 10$

## 输出格式

- 单个整数：表示太空电梯的最高高度

首先按安全高度排序，因为安全高度小的要想被利用，只能呆在最下面。

之后依次添加，只要在安全高度以内，有多少价多少。

# 平分子集

(Subset Sums, 1998 Spring)

奶牛正在学习算术，今天的作业是抄写1到N之间的所有整数，然后找到尽量多的分组方法，把这些整数分成总和相等的两部分。当 $N = 3$ 时，只有一种分法，一边是{1,2}，另一边是{3}。当 $N = 7$ 时，就有四种分法了：

{1,6,7}和{2,3,4,5}，{2,5,7}和{1,3,4,6}，{3,4,7}和{1,2,5,6}，{1,2,4,7}和{3,5,6}

请你帮奶牛计算一下，对于一个给定的N，有多少分法？

## 输入格式

- 单个整数：N， $1 \leq N \leq 50$

## 输出格式

- 单个整数：表示方案数，如果连一种分法也不存在，输出 0，保证答案小于 $2^{63}$

```
for(int i=1;i<=N;i++)
```

```
    for(int j=N*(N+1)/4;j>=i;j--)//01 背包
```

```
        dp[j]+=dp[j-i];
```

```
cout<<dp[N*(N+1)/4]/2;
```

```
//二分之总数的情况除以 2 是因为两部分对称
```

# 提交作业

## (Turning in Homework, 2004 Open)

贝西在哞哞大学选修了C门课，她要把这些课的作业交给老师，然后去车站和同学们一起回家。老师们在办公室里，办公室要等他们下课后才开，第i门课的办公室在 $T_i$ 时刻后开放。

所有的办公室都在一条走廊上，这条走廊长H米，一开始贝西在走廊的最左边，第i门课的办公室距离贝西的长度为 $X_i$ ，车站距离贝西的长度为B。

贝西可在走廊上自由行走，每时刻可以向右或者向左移动一单位的距离，也可以选择在任何地方暂停。贝西如果走到办公室所处的位置，而且这间办公室已经开门了的话，就可以把作业交掉，不用花时间在走进办公室上。

请帮助贝西确定交完所有作业，再走到车站的最短时间。

由大区间推出小区间。

$F[i][j][0]$ 表示在整个  $i \sim j$  区间里只有  $i$  这个作业交了；

$F[i][j][1]$ 表示在整个  $i \sim j$  区间里只有  $j$  这个作业交了；

初始化：

$dp[1][n][0] = \max(a[1].x, a[1].t)$  在  $1 \sim n$  区间内只交了 1 作业

$dp[1][n][1] = \max(a[n].x, a[n].t)$  在  $1 \sim n$  区间内只交了  $n$  作业

状态转移方程：

$$dp[i][j][0] = \min(\max(dp[i-1][j][0] + x[i] - x[i-1], t[i]), \max(dp[i][j+1][1] + x[j+1] - x[i], t[i]))$$
$$dp[i][j][1] = \min(\max(dp[i][j+1][1] + x[j+1] - x[j], t[j]), \max(dp[i-1][j][0] + x[j] - x[i-1], t[j]))$$



# 抢鲜草

(Turning in Homework, 2004 Open)

贝西在一条笔直的马路上，发现她前后到处都是新鲜的青草！一共有 $N$ 棵青草，第 $i$ 棵青草的坐标为 $X_i$ 。贝西一开始在坐标 $K$ 处，每分钟可以向前或向后移动一个单位的距离。不过，青草太新鲜了，每棵青草每分钟会损失一单位的口感。

作为吃货的贝西当然要吃光所有的青草，但她也会在意口感的损失。请你告诉她该怎样来回跑动，才能在口感损失之和最小的情况下消灭所有的青草。贝西的食速很快，不计吞下青草的时间。

## 输入格式

- 第一行：两个整数： $N$ 和 $K$ ， $1 \leq N \leq 1000$ ， $1 \leq K \leq 10^6$
- 第二行到 $N+1$ 行：第 $i+1$ 行有一个整数： $X_i$ ， $1 \leq X_i \leq 10^6$

## 输出格式

- 单个整数：表示最小损失的口感之和

奶牛逃跑型 dp.

$f[i][j][1]$ 表示吃完  $i \sim j$  后停在  $i$  的最小损失， $f[i][j][0]$  表示吃完  $i \sim j$  后停在  $j$  的最小损失。

状态转移方程：

$$\begin{aligned} f[i][j][1] &= \min(f[i-1][j][1] + (N-i-j+1) * (a[i] - a[i-1]), \\ & f[i-1][j][0] + (N-i-j+1) * (a[i] + b[j])); \\ f[i][j][0] &= \min(f[i][j-1][0] + (N-i-j+1) * (b[j] - b[j-1]), \\ & f[i][j-1][1] + (N-i-j+1) * (a[i] + b[j])); \end{aligned}$$

# 最优回文

(Cheapest Palindrome, 2007 Open)

追踪奶牛是一件棘手的任务。约翰安装了一套自动系统，为每头奶牛设置了电子身份标签，当奶牛通过扫描器的时候，系统可以读取奶牛的身份信息。标签是由字符串组成的，长度为M，所用字符由N个小写字母组成。

奶牛是顽皮的动物，她们有事会倒着通过扫描器。这样一来，系统会把 `abcb` 读成 `bcba`，识别发生了障碍。但是 `abcba` 就不会发生这个问题，因为它是一个回文。

于是约翰想把所有的标签都改成回文，比如在 `abcb` 的最后加个 `a` 就变成了 `abcba`；也可以在前面加上 `bc`，变成 `bcbabcb`；或者去除字母 `a`，只保留 `bc`。约翰可以在任意位置删除或插入字符。不巧的是，标签是电子产品，增加或删除字母都要付费。给定一头奶牛的身份标签，以及增加删除相关字母的费用，找出把字符串变成回文的最小费用。注意空字符串也算是一条回文。

## 输入格式

- 第一行：两个整数：N和M， $1 \leq M \leq 2000$ ， $1 \leq N \leq 26$
- 第二行：一个长M的字符串，代表原始的身份标签
- 第三行到第N+2行：每行为一个用空格分开的三元组：其中包括一个字符和两个整数，分别表示增加或删除这个字符的费用，所有的费用满足 $0 \leq \text{费用} \leq 10000$

## 输出格式

- 单个整数：表示改造身份标签的最小费用

$F[i][j]$ 表示将  $i \sim j$  变为回文串的最小代价，显然  $f[i][i]=0$ 。

状态转移方程：

$$\text{if}(a[i-1]==a[j+1])f[i-1][j+1]=f[i][j]$$

$$f[i][j]=\min(f[i-1][j]+v[a[i]],f[i][j-1]+v[a[m-j+1]])$$



# 智取金币

(Treasure Chest, 2010 Dec)

贝西和她的朋友邦妮挖到了一个宝箱，里面藏着 $N$ 枚金币！但是金币对奶牛没用，她们一直用这些金币来玩游戏。游戏开始之前，金币会被排成一条队列，在第 $i$ 个位置的金币价值为 $C_i$ 。贝西和邦妮轮流取金币，轮到拿的时候，只能选择队列的最前面的那块，或队列最后面的一块，所有金币取完之后，游戏就结束了。

贝西和邦妮都是非常聪明的，她们从不犯错，会采用最好的策略让自己取到的金币价值最大。假设贝西是先手，请帮她计算一下，她最多能拿价值多少的金币？

## 输入格式

- 第一行：单个整数： $N$ ， $1 \leq N \leq 5000$
- 第二行到第 $N+1$ 行：第 $i+1$ 行有一个整数： $C_i$ ， $1 \leq C_i \leq 5000$

## 输出格式

- 单个整数：表示如果双方都按最优策略进行游戏，先手可以拿到的最大价值之和

$F[i][j]$ 表示先手在区间  $i \sim j$  的最大值。 $G[i][j]$ 表示后手在区间  $i \sim j$  的最大值。

$$f[i][j] = \max(a[i] + g[i+1][j], a[j] + g[i][j-1])$$

$$g[i][j] = \text{sum}[j] - \text{sum}[i-1] - f[i][j]$$

所以

$$F[i][j] = \text{sum}[j] - \text{sum}[i-1] - \min(f[i+1][j], f[i][j-1])$$

# 打扫食槽

(Cleaning Up, 2009 Mar)

牧场里有 $N$ 头奶牛，约翰向它们提供 $M$ 种食物，第 $i$ 头奶牛吃的是第 $P_i$ 种食物。约翰每天都要打扫食槽，这活很累。奶牛沿着食槽排成一条直线，约翰在打扫时，可以将食槽分割成若干个区间，如果一段区间中有 $K$ 种不同的食物，那么打扫这段区间的时间就是 $K^2$ 。请帮助约翰计划一下，怎样才能使打扫整个食槽的时间最少。

## 输入格式

- 第一行：两个用空格分开的整数： $N$ 和 $M$ ， $1 \leq M \leq N \leq 40000$
- 第二行到 $N + 1$ 行：第 $i + 1$ 行有一个整数 $P_i$ ， $1 \leq P_i \leq M$

## 输出格式

- 单个整数，表示约翰完成打扫的最短时间

显然， $f[i]$ 定义为前 $i$ 头牛所使用的食槽最少时间。

$$F[i] = \min\{f[j] + \text{calc}(j+1, i)\} (j < i)$$

$\text{Calc}$  表示计算区间  $j+1 \sim i$  的不同总类数。要在最开始计算前一次性统计出来(Hash)。

当然， $O(N^2)$ 是对不完的。

# 奶牛自行车

(Cow Cycling, Feb 2002)

有N头奶牛组队参加一个自行车队比赛。比赛在体育场内进行，共需绕场S圈。车队在比赛时排成一列，由于空气阻力的作用，领队奶牛消耗的体力要比后面的多。每头奶牛的初始体力值都是相同的，记作M，体力减为负数的奶牛会在中途掉队，掉队的奶牛不能继续参加比赛，体力不会在比赛途中恢复，但最后只要有一位队员到达终点就算完成了比赛。

比赛最小的单位时间是分钟。车队在每分钟必须绕赛场整数圈，最少一圈。每过一分钟，车队可以自由地更换领队，只要领队奶牛有足够的体力，车队也可以自由地改变车速。在一分钟里，如果车队绕场x圈，领队奶牛将会消耗 $x^2$ 点体力，跟在后面的奶牛将会消耗x点体力，设定速度时，要保证领队奶牛在这一分钟的领队状态下不会掉队。

作为教练，请你计划一下，在比赛进行的每分钟里，应采用什么样的策略才能让车队以最快的时间完成比赛？由于输入数据中保证 $S \leq M$ ，所以一定存在完成比赛的方案。

## 输入格式

- 第一行：三个整数：N，M和S， $1 \leq N \leq 20$ ， $1 \leq S \leq M \leq 100$

## 输出格式

- 单个整数：表示最早完成比赛的时间

显然，每只奶牛只会担任一次领队，卸任后可以舍弃，毕竟只要有一头奶牛到达就行。

$F[i][j][k]$ 表示现在由第i只奶牛领跑，共跑了j圈，i已消耗k体力。

状态转移方程：

$$F[i][j+p][k+p^2] = \min(F[i][j+p][k+p^2], f[i][j][k]+1)$$

$$F[i+1][j][j] = \min(F[i+1][j][j], f[i][j][k])$$

# 滑雪缆车

(Ski Lift, 2006 Mar)

科罗拉多州的罗恩打算为奶牛建造一个滑雪场，为此要在山上规划一条缆车线路。整座山可以看成是二维平面上的一条折线。这条折线可以分为 $N - 1$ 条线段， $N$ 个顶点，第 $i$ 个顶点的横坐标就是 $i$ ，纵坐标为 $H_i$ ，相邻顶点之间的水平距离都是1，其中第一个点称作起点，最后一个点称作终点。

缆车线路必须从起点开始修建，结束于终点。中间可以选择一些拐点安放缆绳的支柱，安全标准有两点：

第一，缆绳的跨度有限制——相邻支柱的水平距离不能超过 $K$ ；

第二，缆绳的高度有限制——两根支柱之间的钢丝视作笔直的，在这座山的任何位置，钢丝都不能低于山的高度，但允许缆绳紧贴在山坡上或恰好穿过某个山峰。支柱相对于拐点的高度不计。

为了节约，罗恩希望修建的支柱越少越好，请帮他规划一下吧！起点和终点上是一定要修建支柱的。

## 输入格式

- 第一行：两个整数： $N$ 和 $K$ ， $2 \leq N \leq 5000$ ， $1 \leq K \leq N - 1$
- 第二行到 $N + 1$ 行：第 $i + 1$ 行有一个整数： $H_i$ ， $0 \leq H_i \leq 10^9$

## 输出格式

- 单个整数：表示最少需要修建的支柱数量

显然， $f[i]$ 表示前 $i$ 座山修的最少支柱。

$$F[i] = f[j] + 1 (j \text{ 合法})$$

PS:  $j$  合法即：1.  $i - j \leq k$

2.  $i \sim j$  的斜率不能小于  $i, j$  之中的最大斜率，即不能穿越山峰

# 奶牛飞盘队

(Cow Frisbee Team, 2009 Mar)

老唐最近迷上了飞盘，约翰想和他一起玩，于是打算从他家的 $N$ 头奶牛中选出一些组成一支队伍。每只奶牛的能力为整数，第 $i$ 头奶牛的能力为 $R_i$ ，一支队伍的总能力就是所有队员能力的总和。

约翰比较迷信，他的幸运数字是 $F$ ，所以他要求队伍的总能力必须是 $F$ 的倍数。请帮他算一下，符合这个要求的队伍组合有多少？由于这个数字很大，只要输出答案除以 $10^8$ 的余数就可以了。当然队伍里至少要有一头奶牛，没有奶牛不能算一只队伍。

## 输入格式

- 第一行：两个整数： $N$ 和 $F$ ， $1 \leq N \leq 2000$ ， $1 \leq F \leq 1000$
- 第二行到 $N + 1$ 行：第 $i + 1$ 行有一个整数 $R_i$ ，表示第 $i$ 头奶牛的能力， $1 \leq R_i \leq 10^5$

## 输出格式

- 单个整数：表示方案数除以 $10^8$ 的余数

$F[i][j]$ 表示前 $i$ 个数的组合  $\text{mod } F = j$  的方案数。

状态转移方程：

$$F[i][j] += f[i-1][j];$$

$$F[i][(j+a[i])\%F] += f[i-1][j];$$

请脑补答案（及过程） $\text{mod } 10^8$  部分



# 滑雪比赛

(Bobsledding, 2009 Dec)

贝西参加了一场高山急速滑雪比赛，滑道总长度为 $L$ 。出发时，她的初速度为每秒 1 米，贝西可以在比赛过程中加快或减少速度，每过 1 米，她能将速度加 1、减 1 或保持不变。在滑雪的过程中，贝西会遇到 $N$ 个转弯点，编号为 $i$ 的转弯点距离起点有 $T_i$ 米。安全起见，贝西到达 $i$ 号转弯点时速度不能超过 $S_i$ 。穿过终点的速度是没有限制的，请问在整个比赛过程中，贝西能够达到的最大速度是多少？

## 输入格式

- 第一行：两个整数： $L$ 和 $N$ ， $2 \leq L \leq 10^9$ ， $1 \leq N \leq 10^5$
- 第二行到 $N + 1$ 行：第 $i + 1$ 行有两个整数： $T_i$ 和 $S_i$ ， $1 \leq T_i \leq L - 1$ ， $1 \leq S_i \leq 10^9$

## 输出格式

- 单个整数：表示贝西在比赛过程中能够达到的最大速度

$F[i]$ 表示在  $i$  点前能达到的最大速度。

```
for(int i=1;i<=n+1;i++){
    int start_speed=min(f[i-1],s[i-1]); //t0
    int delta_speed=start_speed-s[i]; //Δt
    int distance=t[i]-t[i-1]; //s
    int tmp=distance-abs(delta_speed);
    //是否可以加速到最大
    if(tmp<0) f[i]=start_speed+distance;
    else f[i]=tmp/2+max(s[i],start_speed);
}
```

# 滑雪课程

(Ski Lessons, 2009 Open)

约翰请贝西去科罗拉多去滑雪。不过贝西不太会玩，她只是个滑雪能力为1的渣渣。所以她决心参加一些滑雪课程。滑雪场提供S门课程，第i门课的开始时间是 $M_i$ ，持续时间为 $L_i$ ，上课不能迟到或早退。上完课之后，贝西的滑雪能力将变成 $A_i$ 。注意，不是能力增加 $A_i$ ，而是变成了 $A_i$ 。

滑雪场有N条斜坡，第i条斜坡滑行一次需要 $D_i$ 分钟，要求游客的滑雪能力达到 $C_i$ 或以上时才能进入。

贝西可以随意安排她的时间：滑雪、上课，或美美地喝上一杯可可汁，但她在滑雪场只能呆到第T分钟。请问她如何安排时间，滑行次数才能尽量多？

## 输入格式

- 第一行：三个整数T, S和N,  $1 \leq T \leq 10^4$ ,  $1 \leq S \leq 100$ ,  $1 \leq N \leq 10^5$
- 第二行到S+1行：第i+1行描述了第i门课程，分别为 $M_i$ ,  $L_i$ 和 $A_i$ ,  $1 \leq M_i, L_i \leq 10^4$ ,  $1 \leq A_i \leq 100$
- 第S+2行到S+N+1行：第S+i+1行描述了第i条斜坡，分别为 $C_i$ 和 $D_i$ ,  $1 \leq C_i \leq 100$ ,  $1 \leq D_i \leq 10^4$

## 输出格式

- 第一行：单个整数，表示在时限内贝西可以滑完的最大次数

```
for(int i=1;i<=t;i++){
    for(int j=1;j<=100;j++){
        f[i][j]=f[i-1][j];
        if(ke[i-1][j])f[i][j]=max(f[i][j],g[ke[i-1][j]]);
        if(i-po[j]>=0)f[i][j]=max(f[i][j],f[i-po[j]][j]+1);
        g[i]=max(g[i],f[i][j]);
    }
}
```



# 方形牛棚

(Big Barn, 1997 Fall)

约翰有一片很大的正方形农场,可均匀地分为 $N \times N$ 个方格,他打算从中选择一片区域,建造一座最大的牛棚。牛棚也必须是正方形的,长宽相等,且它的边界必须平行于农场的边界。这片农场中有些地方被繁茂的森林占据了,约翰的牛棚必须绕开这些地方。请你帮助约翰找到符合这个条件的最大的方形牛棚吧。例如下面这个例子中,#代表森林:

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | . | . | . | . | . | . | . | . |
| 2 | . | # | . | . | . | # | . | . |
| 3 | . | . | . | . | . | . | . | . |
| 4 | . | . | . | . | . | . | . | . |
| 5 | . | . | . | . | . | . | . | . |
| 6 | . | . | # | . | . | . | . | . |
| 7 | . | . | . | . | . | . | . | . |
| 8 | . | . | . | . | . | . | . | . |

那么最大的牛棚存在于右下角,大小为 $5 \times 5$

## 输入格式

- 第一行: 两个整数:  $N$ 和 $M$ ,  $1 \leq N \leq 1000$ ,  $1 \leq M \leq 10000$
- 第二行到第 $M+1$ 行: 每行有两个整数 $X$ 和 $Y$ , 表示一个被树林占据的格子的坐标,  $1 \leq X \leq N$ ,  $1 \leq Y \leq N$

## 输出格式

- 单个整数: 表示牛棚的最大边长

$F[i][j]$ 是以  $i, j$  为右上角的最大正方形。

$\text{if}(a[i][j] == \text{'\#'}) f[i][j] = 0;$

$\text{else } F[i][j] = \min(f[i-1][j], f[i][j-1], f[i-1][j-1]) + 1$

# 接住苹果

(Apple Catching, 2004 Nov)

奶牛喜欢吃苹果。约翰有两棵苹果树，有 $N$ 只苹果会从树上陆续落下。如果站在掉落苹果的那棵树下，贝西就能接住苹果。贝西一开始在第一棵树下。在苹果掉落的过程中，她可以来回走动，但她的运动能力不足，最多只能移动 $K$ 次。请计算一下她最多可以接住几只苹果。

## 输入格式

- 第一行：两个整数： $N$ 和 $K$ ， $1 \leq N \leq 1000$ ， $1 \leq K \leq 30$
- 第二行到第 $T + 1$ 行：第 $i + 1$ 行有一个整数 $T_i$ ，表示第 $i$ 只苹果从哪棵树上掉落，1 表示从第一棵树，2 表示从第二棵树

## 输出格式

- 单个整数：表示能接住的最大苹果数量

$F[i][j]$ 表示第 $i$ 时间，已经移动了 $j$ 次所取得的最大值。

状态转移方程：

$$\begin{aligned} F[i][j] = & f[i-1][j] + a[(i-1)\%2][i-1] \\ & + f[i-1][j-1] + a[i\%2][i-1] \end{aligned}$$

# 公司利润

## (Profits, 2011 Jan)

奶牛开了家新公司，已经连续运作了 $N$ 天。她们在第 $i$ 天获得了 $P_i$ 元的利润，不过有些天的利润可能是负的，表示亏损。约翰想给新公司写个大新闻，以吹嘘她们的业绩。请你帮助他选出一段连续的日子，使得新公司在这段日子中的利润之和是最大的。这段日子的长度不限，但至少应包括一天。

### 输入格式

- 第一行：单个整数： $N$ ， $1 \leq N \leq 10^5$
- 第二行到 $N + 1$ 行：第 $i + 1$ 行有一个整数： $P_i$ ， $-1000 \leq P_i \leq 1000$

### 输出格式

- 单个整数，表示一段连续日子中最大的利润之和

最大连续子段和。

$$F[i] = \max(f[i-1], 0) + a[i]$$