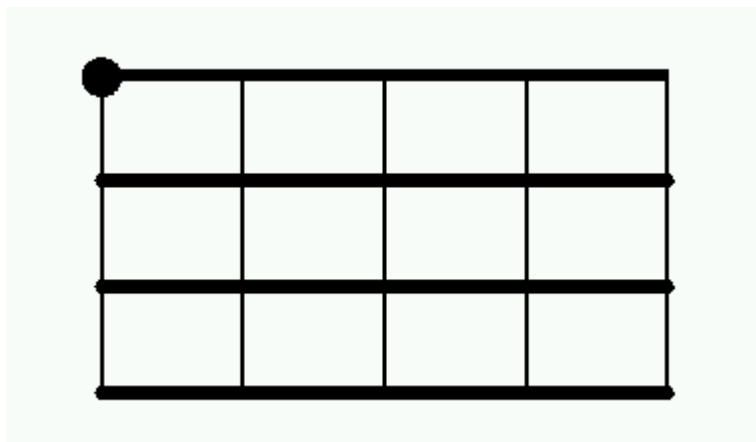


Translate:USACO/vans

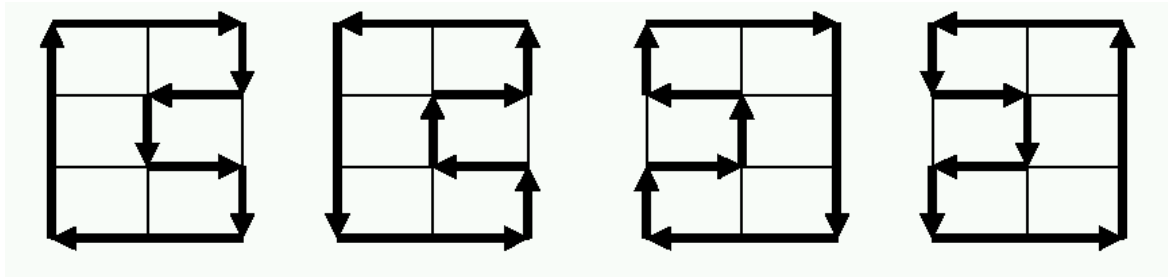
补充翻译 by tjbwyk

描述

如  $N=5$  时，郊区如下图所示，圆点表示邮局，直线表示街道。



另一个例子，下面四幅图表示了当  $N=3$  时的全部四种情况



## 格式

**PROGRAM NAME:** vans

**INPUT FORMAT:**

(file vans.in)

INPUT FORMAT 一行：一个数值 N

**OUTPUT FORMAT:**

(file vans.out) 一行：到 INPUT 中给出的街道的路径总数

## SAMPLE INPUT

4

## SAMPLE OUTPUT

12

## 解题报告：

找规律。第一张图和第二张图其实已经暗示了这道题目的潜在规律。注意到从一个第一行的点横向出发再回到这个点只有从这个点下面的点回来。定义  $F[i]$  表示从前  $i$  列中第  $i$  列的第 2 个点到第 1 个点共有多少合法路径，根据这个定义  $F[n]$  即为所求。 $G[i]$  表示前  $i$  列中第  $i$  列的第 1 个点到第 4 个点共有多少路径。因此可以推出  $F[i] = F[i-1] + G[i-1]$  (画图找规律)。

下面我们仔细观察这两个方程的定义，从而化简方程，如下图：

$$G[i] = F[i-1]*2 + G[i-2] + G4[i] \quad (1)$$

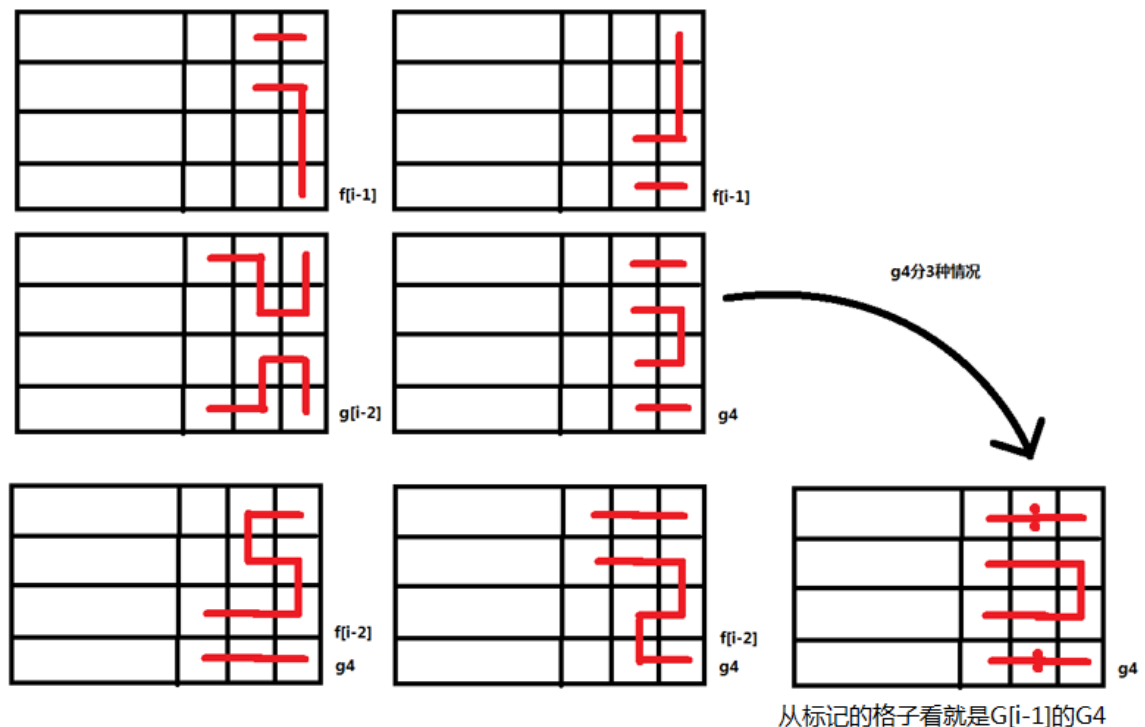
$$G4[i] = F[i-2]*2 + G4[i-1] \quad (2)$$

$$\text{由(1)化简(2)式得到: } G4[i] = G[i-1] - G1[i-1] - G2[i-1] - G3[i-1] + F[i-2] * 2 = G[i-1] - G[i-3]$$

$$\text{所以 } G[i] = F[i-1]*2 + G[i-1] + G[i-2] - G[i-3] \quad (3)$$

$$\text{又由 } F[i] - F[i-1] = G[i-1] \quad (4)$$

最终我们得到  $F[n] = 2F[n-1] + 2F[n-2] - 2F[n-3] + F[n-4]$  ( $n \geq 5$ )  
 初始值:  $F[1] = 0$ ;  $F[2] = 2$ ;  $F[3] = 4$ ;  $F[4] = 12$ 。



## Translate:USACO/rectbarn

Mircea Pasoi -- 2003

译 by yangzhe1990

### 描述

到底是个资本家，Farmer John 想通过买更多的奶牛来扩大它的生意。它需要给奶牛建造一个新的牛棚。FJ 买了一个矩形的  $R(1 \leq R \leq 3000)$  行  $C(1 \leq C \leq 3000)$  列的牧场。不幸的是，他发现某些  $1 \times 1$  的区域被损坏了，所以它不可能在把整个牧场建造成牛棚了。FJ 数了一下，发现有  $P(1 \leq p \leq 30000)$  个  $1 \times 1$  的损坏区域并且请你帮助他找到不包含损坏区域的面积最大的矩形的牛棚。

- 对同一个损坏区域可能有多次描述

题目名称: rectbarn

### 输入格式 (file rectbarn.in)

第 1 行: 三个空格隔开的整数  $R, C$ , and  $P$ .

第 2..P+1 行: 每行包含两个空格隔开的整数, r 和 c, 给出一个损坏区域的行号和列号.

## 输出格式 (file rectbarn.out)

第 1 行: 牛棚的最大可能面积

## 样例输入

```
3 4 2
1 3
2 1
```

## 样例输出

```
6
```

## 输出解释

```
  1 2 3 4
. +--+--+--+
1 |  | |X|  |
. +--+--+--+
2 |X|#|#|#|
. +--+--+--+
3 | |#|#|#|
. +--+--+--+
```

标 'X' 的区域是损坏的, 标 '#' 的区域是牛棚.

## 解题报告:

这道题目乍一看很类似 5.3.4 但是因为这道题目是矩形而不是正方形, 所以要考虑到转移方程会出现多种不同的状况。但是鉴于数据规模我们还是要使用 DP, 主要思想是极大化思想, 如果一个子矩阵向左向右都不能扩展, 那么这个子矩阵就是极大子矩阵。显然我们要找到的答案肯定是一个极大子矩阵。

设  $H[i][j]$  表示点  $(i, j)$  所能向上伸展的最大高度, 那么有  $H[i][j] = H[i-1][j] + 1$  (点  $i, j$  不是障碍)

$left[i][j]$  表示点  $i, j$  向上长度为  $H[i][j]$  的线段向左能扩展的最长距离, 同理  $right[i][j]$  表示点  $i, j$  向上长度为  $H[i][j]$  的线段向右能扩展的最长距离,  $left[i][j] = \min(left[i-1][j], maxleftlen)$ ,  $maxleftlen$  是点  $(i, j)$  向左伸展的最长距离。 $right$  的转移方程同理。也就是这道题目与 5.3.4 不同之处

之一是要考虑向右扩展，这是因为 left 是基于 H 的，如果只考虑向左的话我们的结果只能保证矩阵尽量的高，他横向的宽度不能保证，这个时候同时向右扩展也就是解决了宽度的问题。毕竟矩阵左右两边都应该是一个极大的界。

## Translate:USACO/cowxor

---

**Cow XOR** 奶牛异或 Adrian Vladu -- 2005

译 by yangzhe1990

---

### 描述

---

农民约翰在喂奶牛的时候被另一个问题卡住了。他的所有  $N(1 \leq N \leq 100,000)$  个奶牛在他面前排成一行(按序号  $1..N$  的顺序)，按照它们的社会等级排序。奶牛 #1 有最高的社会等级，奶牛 #N 最低。每个奶牛同时被指定了一个不唯一的附加值，这个数在  $0..2^{21} - 1$  的范围内。

帮助农民约翰找出应该从哪一头奶牛开始喂，使得从这头奶牛开始的一个连续子序列上，奶牛的附加值的异或最大。

如果有多个这样的子序列，选择结尾的奶牛社会等级最高的。如果还不唯一，选择最短的。

### 格式

---

**PROGRAM NAME:** cowxor

**INPUT FORMAT:**

(file cowxor.in)

**INPUT FORMAT**

第 1 行: 一个单独的整数  $N$ 。

第 2 到  $N + 1$  行:  $N$  个  $0..2^{21} - 1$  之间的整数，代表每头奶牛的被赋予的数。第  $j$  行描述了社会等级  $j - 1$  的奶牛。

**OUTPUT FORMAT:**

(file cowxor.out)

第 1 行: 3 个空格隔开的整数，分别为：最大的异或值，序列的起始位置、终止位置。 时限 0.5 秒

### SAMPLE INPUT

---

5

```
1
0
5
4
2
```

## SAMPLE OUTPUT

---

```
6 4 5
```

## 样例输出说明

---

最大异或值为 6，从第 4 个开始喂，到第 5 个结束。

4 异或 2 = 6

(100) 异或 (010) = (110)

## 解题报告：

异或的逆运算还是异或。所以先处理总和的值， $\text{num}[i]$ 表示前  $i$  个牛的异或值。所以从  $i$  到  $j$  的异或值就是  $\text{num}[j] \text{ xor } \text{num}[i-1]$ 。但是具体查询不能朴素，时间复杂度是  $O(n^2)$  会超时。考虑到异或的性质，越是高位不同的话值会越大，所以我们贪心寻找即可，数据结构使用 0-1 字典树存储，每次都贪心寻找与当前的异或各位尽可能不同，且不同位尽可能高的值，最后出现的结果就是最优的。