

# SN 省队训练 1st

题目名称	跳	土地划分	最大异或和
输入文件名	jump.in	divide.in	xor.in
输出文件名	jump.out	divide.out	xor.out
每个测试点时限	2s	1s	3s
内存限制	512 MB	512 MB	512 MB
测试点数目	20	10	10
每个测试点分值	5	10	10
是否有部分分	无	无	无
题目类型	传统型	传统型	传统型

提交源程序须加后缀

对于 Pascal 语言	.pas	.pas	.pas
对于 C 语言	.c	.c	.c
对于 C++ 语言	.cpp	.cpp	.cpp

注意：最终测试时，所有编译命令均不打开任何优化开关。

# 跳 (jump)

## 【问题描述】

邪教喜欢在各种各样空间内跳。

现在，邪教来到了一个二维平面。在这个平面内，如果邪教当前跳到了 $(x,y)$ ，那么他下一步可以选择跳到以下 4 个点： $(x-1,y)$ ,  $(x+1,y)$ ,  $(x,y-1)$ ,  $(x,y+1)$ 。

而每当邪教到达一个点，他需要耗费一些体力，假设到达 $(x,y)$ 需要耗费的体力用  $C(x,y)$ 表示。

对于  $C(x,y)$ ，有以下几个性质：

- 1、若  $x=0$  或者  $y=0$ ，则  $C(x,y)=1$ 。
- 2、若  $x>0$  且  $y>0$ ，则  $C(x,y)=C(x,y-1)+C(x-1,y)$ 。
- 3、若  $x<0$  且  $y<0$ ，则  $C(x,y)=$ 无穷大。

现在，邪教想知道从 $(0,0)$ 出发到 $(N,M)$ ，最少花费多少体力（到达 $(0,0)$ 点花费的体力也需要被算入）。

由于答案可能很大，只需要输出答案对  $10^9+7$  取模的结果。

## 【输入格式】(jump.in)

读入两个整数  $N$ ,  $M$ ，表示邪教想到达的点。

## 【输出格式】(jump.out)

输出仅一个整数，表示邪教需要花费的最小体力对  $10^9+7$  取模的结果。

## 【样例输入】

1 2

## 【样例输出】

6

## 【数据说明】

对于 10%的数据，满足  $N, M \leq 20$ ；

对于 30%的数据，满足  $N, M \leq 100$ ；

对于 60%的数据，满足  $\min(N,M) \leq 100$ ；

对于 100%的数据，满足  $0 \leq N, M \leq 10^{12}$ ,  $N * M \leq 10^{12}$ 。

# 土地划分 (divide)

## 【问题描述】

Y 国有  $N$  座城市，并且有  $M$  条双向公路将这些城市连接起来，并且任意两个城市至少有一条路径可以互达。

Y 国的国王去世之后，他的两个儿子 A 和 B 都想成为新的国王，但他们都想让这个国家更加安定，不会用武力解决问题。

于是他们想将这个国家分成两个小国家 A 国和 B 国。现在，A 拥有 1 号城市，B 拥有  $N$  号城市，其他的城市还尚未确定归属哪边（划分之后的国家内部城市可以不连通）。

由于大家都想让国家变得更好，而某些城市的人民愿意国王的 A 儿子作为他们的领袖，而某些城市更看好 B，而为了交通的便捷，如果划分后的公路连接两个同一个国家的城市，那么更利于城市之间的交流。于是大臣们设计了一种对土地划分的评分机制，具体如下：

- 1、对于城市  $i$ ，如果它划分给 A 国，将得到  $VA[i]$  的得分；划分给 B 国，将得到  $VB[i]$  的得分。
- 2、对于一条公路  $i$ ，如果它连接两个 A 国的城市，将得到  $EA[i]$  的得分；连接两个 B 国的城市，将得到  $EB[i]$  的得分；否则，这条公路将失去意义，将扣除  $EC[i]$  的得分。

现请你找到最优的土地划分，使得这种它的评分最高。

## 【输入格式】 (divide.in)

第一行包含两个整数  $N, M$ ，含义如问题描述所示。

接下来一行  $N-2$  个非负整数，表示  $VA[2..N-1]$ 。

接下来一行  $N-2$  个非负整数，表示  $VB[2..N-1]$ 。

接下来  $M$  行，每行五个非负整数描述一条公路：X Y  $EA[i]$   $EB[i]$   $EC[i]$ ，含义如问题描述所示。

## 【输出格式】 (divide.out)

输出有且仅有一个整数，表示最高评分。

## 【样例输入】

```
3 3
8
9
1 2 2 6 2
2 3 8 5 7
1 3 9 4 1
```

**【样例输出】**

11

**【样例说明】**

A 国仅有 1 号点，B 国有 2 号和 3 号点。

评分=VB[2]+EB[2]-EC[1]-EC[3]=9+5-2-1=11。

**【数据说明】**

数据点	N	M	备注
1-2	$\leq 20$	$\leq 200$	无
3-4	$\leq 5000$	$\leq 10000$	VA、VB、EA、EB 均为 0
5-6	$\leq 5000$	$\leq 10000$	EC 均为 0
7-10	$\leq 10000$	$\leq 40000$	无

# 最大异或和 (xor)

## 【问题描述】

给定一个非负整数序列 $\{a\}$ ，初始长度为  $N$ 。

有  $M$  个操作，有以下两种操作类型：

- 1、 $A\ x$ ：添加操作，表示在序列末尾添加一个数  $x$ ，序列的长度  $N+1$ 。
- 2、 $Q\ l\ r\ x$ ：询问操作，你需要找到一个位置  $p$ ，满足  $l \leq p \leq r$ ，使得：  
 $a[p] \text{ xor } a[p+1] \text{ xor } \dots \text{ xor } a[N] \text{ xor } x$  最大，输出最大是多少。

## 【输入格式】(xor.in)

第一行包含两个整数  $N, M$ ，含义如问题描述所示。

第二行包含  $N$  个非负整数，表示初始的序列  $A$ 。

接下来  $M$  行，每行描述一个操作，格式如题面所述。

## 【输出格式】(xor.out)

假设询问操作有  $T$  个，则输出应该有  $T$  行，每行一个整数表示询问的答案。

## 【样例输入】

```
5 5
2 6 4 3 6
A 1
Q 3 5 4
A 4
Q 5 7 0
Q 3 6 6
```

## 【样例输出】

```
4
5
6
```

## 【数据说明】

本题共有 10 个测试点，每个测试点 10 分。

对于测试点 1-2， $N, M \leq 5$ 。

对于测试点 3-7， $N, M \leq 80000$ 。

对于测试点 8-10， $N, M \leq 300000$ 。

其中测试点 1, 3, 5, 7, 9 保证没有修改操作。

对于 100% 的数据， $0 \leq a[i] \leq 10^7$ 。