

NMHC Round

题外话：I got Uagg

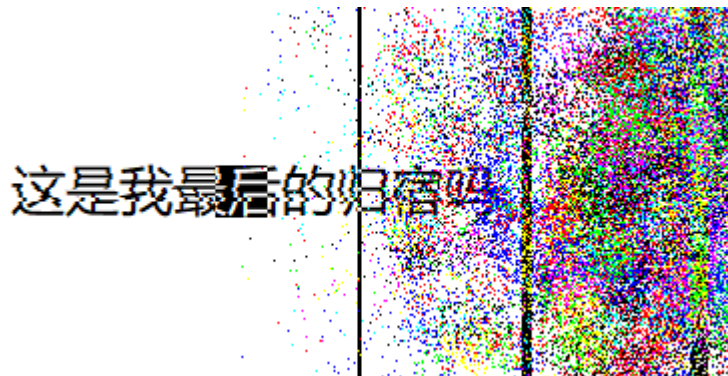
	A	B	C	D
题目名称	计算乘积	Casino in Concrete Math Club	Simple Problem Inversed	古老的金字塔
英文名	product	wheel	simple	pyramid
题目类型	传统题	交互题	传统题	传统题
源代码文件名	product.cpp	wheel.cpp	simple.cpp	pyramid.cpp
输入文件名	product.in	-	simple.in	pyramid.in
输出文件名	product.out	-	simple.out	pyramid.out
测试点数目	5	4	6	10
测试点是否等分	否	否	否	是
时间限制	700ms	1s	1s	300ms
空间限制	256MB	64MB	128MB	64MB

注意事项：

- 除 B 题外，本场考试所有题目都需要从输入文件中读入数据、并输出到输出文件中。
- 编译器的编译参数为 `-lm -std=c++14 -O2 -w1,--stack=2147483647`。
- 具体时间限制可能会根据机房评测机速度进行调整。
- 请注意 B 题的题目名称与英文名的无关性（

A. 计算乘积

题目背景



T1 当然是要来个数列题的啊~

题目描述

给定一个长度为 n 的序列和一个固定的长度 m 。

Q 希望求出这个序列所有长度为 m 的区间的乘积。由于值很大，Q 只想得到答案模 p 的值。

帮助 Q 求出答案，找回他走远的记忆。

输入格式

第一行四个整数 n, m, p, t 。 t 表示输入和输出的方式。

若 $t = 0$ ：

- 第二行 n 个整数表示 a_1, a_2, \dots, a_n 。

若 $t = 1$ ：

- 第二行四个整数 A, B, x, y, c ，表示 $a_1 = A, a_2 = B$ ，且对于 $i > 2$ 有 $a_i = (x \cdot a_{i-1} + y \cdot a_{i-2} + c \cdot i) \bmod p$ 。

数据保证若 $n \geq 2 \times 10^6$ ，则 $t = 1$ 。

输出格式

若 $t = 0$ ，输出一行 $(n - m + 1)$ 个整数，第 i 个整数表示区间 $[i, i + m - 1]$ 的答案 b_i ；否则输出一行一个整数 v 。 v 的值定义如下：

$$v = \bigoplus_{i=1}^{n-m+1} i \cdot b_i$$

样例 1

输入

```
4 2 10 0
1 3 2 5
```

输出

```
3 6 0
```

样例 2

输入

```
5 3 1000000007 0
5423874 2874623 0 21675312 439875233
```

输出

```
0 0 0
```

样例 3

输入

```
5 2 1000000007 1
1 2 1 3 1
```

输出

```
3242
```

样例 4

见下发文件中的 product/ex_product1.in 和 product/ex_product1.out。该样例满足 p 是质数。

样例 5

见下发文件中的 product/ex_product2.in 和 product/ex_product2.out。

提示

【样例 3 提示】

样例 3 对应的数列是 1, 2, 8, 18, 47。

本题开启子任务评测。太好骗分子实在卡不掉啊啊啊造出来的数据都是直接猜就能过的改了造数据参数还是一样反正就十个点可怜可怜我吧

对于 100% 的数据, $1 \leq m \leq n \leq 3 \times 10^6, 10^9 \leq p \leq 2 \times 10^9, 0 \leq a_i < p$ 。

子任务编号	$n \leq$	$t =$	特殊性质	分值
1	2000	0 或 1	无	15
2	3×10^6	1	p 为质数	5
3	3×10^5	0	无	20
4	3×10^6	1	$m > n/2$	20
5	3×10^6	0 或 1	无	40

B. Casino in Concrete Math Club

题目背景

The *Concrete Math Club* has a casino (open only to purchasers of this book) in which there's a roulette wheel with one thousand slots, numbered 1 to 1000.

— Graham, Knuth, and Patashnik [161]

题目描述

这是一道交互题。请在你的程序开头加入 `include "wheel.h"` 语句。

有一个巨大的轮盘，共有 n 个格子，格子从 1 到 n 乱序编号。你想要猜出 n ，因此你会问主持人一些问题。

轮盘上有一个指针，初始指着编号 x 。你可以向主持人进行最多 m 次操作，每次操作可以要求主持人将指针顺时针 / 逆时针旋转 k 个格子。主持人会告诉你旋转后指针指着的编号。

请猜出 n 的大小。如果你成功了，Q 会送给你一个 Common Ant Egg 作为感谢。据说能孵出一只 ~~tea~~

交互格式

你不需要，也不应实现主函数。你需要实现下面的函数：

`int guess(int x, int m)`， x 表示初始指针的位置， m 表示操作次数上限。函数的返回值是你猜出的 n 的大小。交互库恰好会调用一次 `guess` 函数。

你可以调用下面的函数：

`int rotate(bool way, int k)`。`way` 表示旋转的方向，其值为 `False` 表示逆时针旋转 k 格，`True` 表示顺时针旋转 k 格，返回值为操作完成时指针指向的格子编号。**你需要保证 $0 < k \leq 10^9$ ，否则会得到 WA 的评测结果。**你最多只能调用 `rotate` 函数 m 次。保证在评测机环境下，调用 10^6 次 `rotate` 函数的时间小于等于 40ms。

交互库不是自适应的。也就是说， n 和每个格子上的数在交互前就已经确定。

提示

交互库使用方法

我们在下发文件中发送了 `grader.cpp` 和 `wheel.h` 辅助你进行调试。交互库使用方法如下：

- 将你的代码文件 `wheel.cpp` 与上述两个文件置于同一目录下，然后在该目录终端中输入命令 `g++ wheel.cpp grader.cpp -o wheel.exe -std=c++14 -O2`，得到编译后的 `wheel.exe` 文件；
- 若想运行，请在该目录终端中运行 `wheel.exe` 来进行输入输出。如有必要可以重定向输入输出流到文件。

此交互库与真实评测时使用的交互库不同，你的程序也不应该尝试调用交互库内的变量或其他函数。

数据范围

本题开启子任务评测。交互题开个子任务没啥问题吧谁想把错误的做法放过去啊

对于 100% 的数据, $n \leq 10^6$, $m \geq 1000$ 。

子任务编号	$n \leq$	$m =$	特殊性质	分值
1	10^6	10^6	无	10
2	10^6	2000	无	60
3	10^6	1000	有	10
4	10^6	1000	无	20

特殊性质：顺时针第 i 格位置上的号码 $p_i = i$ 。

评分方式

- 如果你猜测的 n 是错误的, 得 0 分;
- 否则, 如果你的询问次数 $c \leq m$, 得满分, 否则得 0 分。

本题样例见下发文件。

C. Simple Problem Inversed

题目背景

writer : \emptyset

Q 刚学 OI 的时候做过这么一道问题：

给定一棵树，第 i 条边上有一个权值 a_i 。定义一个路径的价值为该路径上所有边权的异或和。多次询问某条路径的价值。

Q 回想起当年的时光。

「真是恍若隔世了呢」，他想。

题目描述

然而你终究不是 Q，现在你需要做这么一道题：

给定一棵有 n 个点的树，第 i 条边上需要被赋予一个非负权值 a_i 。定义一个路径的价值为该路径上所有边权的异或和。有 q 条限制，每条限制给定一段路径的起点、终点 s, t 还有该段路径的价值 w ，表示限制路径 $s \rightarrow t$ 的价值是 w 。

判断是否有解。若有解，还需构造一组满足上述 q 条限制的边权 a_1, a_2, \dots, a_{n-1} 使得 $a_1 \oplus a_2 \oplus \dots \oplus a_{n-1}$ 最小。若有多组边权满足条件，只需输出一种构造。

输入格式

第一行输入两个正整数 n, q 表示点数和限制数量。

接下来 $n - 1$ 行每行输入两个正整数 u, v 表示树上的一条边。

接下来 q 行，每行三个整数 s, t, w 表示一个限制。

输出格式

第一行输出一个字符串 Yes 或 No 表示是否有解。

如果有解，第二行输出 $n - 1$ 个非负整数 a_1, a_2, \dots, a_{n-1} 表示你的构造，按照边的输入顺序输出。

样例 1

输入

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

输出

No

样例 2

输入

6 2
1 2
2 3
3 4
2 5
5 6
1 4 2
2 6 7

输出

Yes
4 2 4 1 6

样例 3

见下发文件中的 simple/ex_simple1.in 和 simple/ex_simple1.ans。此样例满足子任务 1 的约束。

样例 4

见下发文件中的 simple/ex_simple2.in 和 simple/ex_simple2.ans。此样例满足子任务 2 的约束。

样例 5

见下发文件中的 simple/ex_simple3.in 和 simple/ex_simple3.ans。此样例满足子任务 3 的约束。

样例 6

见下发文件中的 simple/ex_simple4.in 和 simple/ex_simple4.ans。此样例满足子任务 4 的约束。

样例 7

见下发文件中的 simple/ex_simple5.in 和 simple/ex_simple5.ans。此样例满足子任务 5 的约束。

样例 8

见下发文件中的 simple/ex_simple6.in 和 simple/ex_simple6.ans。此样例满足子任务 6 的约束。

提示

本题开启子任务评测。呜呜呜我对不起你们啊又开了任务

对于 100% 的数据, $1 \leq n \leq 5 \times 10^5$, $0 \leq q \leq 5 \times 10^5$, $0 \leq w < 2^{30}$, $s \neq t$ 。

子任务编号	$n, q \leq$	特殊性质	分值
1	3000	保证 s, t 距离不超过 1	10
\emptyset	5×10^5	保证 $q = 0$	0
2	3000	无	15
3	10^5	对于第 i 条边, $u_i = i, v_i = i + 1$	15
4	10^5	对于第 i 条边, $u_i = 1, v_i = i + 1$	10
5	10^5	无	15
6	5×10^5	最难做	35

本题评分方式如下：

- 如果你没有正确判断是否有解，得 0 分；
- 如果你正确判断了是否有解，但是输出的解无法满足该测试点的 q 条附加限制，得 0 分；
- 否则我们认为你的输出是合法的，并且能得到该测试点 40% 的分数。如果在此基础上你的构造能够最小化 $a_1 \oplus a_2 \oplus \cdots \oplus a_{n-1}$ ，那么可以得到剩余的 60% 分数。
- 一个子任务的得分为该子任务内所有测试点得分的最小值。

另外，我们在下发文件中下发了本题的 Special Judge Checker，用于检测你构造方案的**合法性**，其中合法与【评分方式】中的定义相同。注意本 Checker 与真实评测时的 Checker 不同，不会检查你的答案是否最小化了边权异或和。

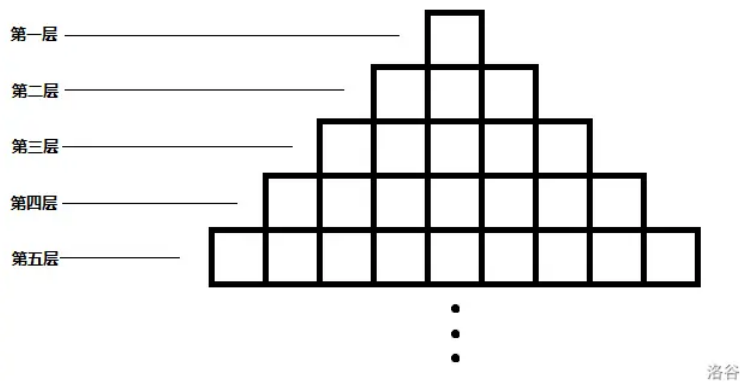
Checker 的使用方式：在对应目录下打开终端，输入命令 `checker [inout file name] [output file name] [answer file name]`，其中 `[input file name]` 是输入文件名称，如 `simple.in`；`[output file name]` 是输出文件名称，如 `simple.out`；`[answer file name]` 是样例答案文件名称，如 `ex_simple1.ans`。

D. 古老的金字塔

题目描述

管理员注：本题为远古题，根据洛谷评测机效率的增长速度以及用户反馈，故将此题的时间限制从原来的 1s 改为 0.3s。

探险家 Q 经过不懈的努力，终于找到了藏有珍贵宝藏的金字塔！然而为了得到宝藏，Q 需要解开守护金字塔大门的密码。金字塔有 N 层，从上往下数第一层有一个石块，第二层有三个石块……以此类推，第 N 层有 $(2N - 1)$ 个石块，如图所示。



在第 N 层的石块上刻着一些数字，从左往右数分别刻着 $A_1, A_2, \dots, A_{2N-1}$ 。在第一层到第 $N - 1$ 层上也刻着数字，每个石块的数字等于它正下方、左下方、右下方三个数字的中位数。而金字塔入口的密码就是第一层上石块的数字。Q 找到了擅长编程的你，希望你帮他求出密码的数值。

注：在本题中，长度为 M 的数组 P_1, P_2, \dots, P_M 的中位数定义为 P 数组中第 $\lceil \frac{M+1}{2} \rceil$ 小的数。

输入格式

- 第一行， N 。
- 第二行， $(2N - 1)$ 个数字，表示 $A_1, A_2, \dots, A_{2N-1}$ 。

输出格式

最终的答案。

样例 1

输入

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

输出

```
4
```

样例 2

输入

```
2
1 2 3
```

输出

```
2
```

样例 3

见下发文件中的 pyramid/ex_pyramid1.in 和 pyramid/ex_pyramid1.ans。该样例满足 $N \leq 1000$ 和特殊性质 A。

样例 4

见下发文件中的 pyramid/ex_pyramid2.in 和 pyramid/ex_pyramid2.ans。该样例满足特殊性质 B。

样例 5

见下发文件中的 pyramid/ex_pyramid3.in 和 pyramid/ex_pyramid3.ans。该样例满足 $N \leq 100000$ 。

提示

$N \leq 100000$ ，所有 $A_i \leq 10^9$ 。

具体数据范围见下表所示。

测试点编号	$N \leq$	$A_i \leq$	特殊性质
1, 2	1000	1999	A
3	100000	10^9	B
4, 5	50000	100	无
6, 7	100000	199999	A
8, 9, 10	100000	10^9	无

特殊性质 A: A 是一个长度为 $(2N - 1)$ 的排列。

特殊性质 B: A 的所有元素均相同。