CQYC Contest Round 3

时间: 11 月 22 日 8:00 - 12:30

题目名称	橡子	气球	蛋糕	基因
题目类型	传统型	传统型	传统型	传统型
可执行文件名	`acorn`	`balloon`	`cake`	`dna`
输入文件名	`acorn.in`	`balloon.in`	`cake.in`	`dna.in`
输出文件名	`acorn.out`	`balloon.out`	`cake.out`	`dna.out`
每个测试点时限	1.0 s	2.5 s	1.5 s	3.0 s
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
子任务 / 测试点数目	4	5	20	8
子任务 / 测试点是否等分	否	否	是	否

提交源程序文件名

对于 C++ 语言	`acorn.cpp`	`balloon.cpp`	`cake.cpp`	`dna.cpp`

编译选项

对于 C++ 语言	`-O2 -lm -std=c++17`

注意事项

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中 `main` 函数的返回值类型必须是 `int`，程序正常返回时的返回值必须是 0。
3. 选手提交的程序代码文件请**直接放在个人目录下**，**不需要**建立子文件夹。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 评测时采用的机器配置为：Intel(R) Core(TM) i5-10500 CPU @3.10GHz，内存 8GB。上述时限以此为准。
8. 评测在当前最新公布的 NOI Linux 下进行，使用 LemonLime 进行评测；各语言的编译器版本以此为准。
9. 本场比赛题目较简单，且难度分布**较为平缓**；故建议阅读所有题目，根据个人情况选择合适的解题策略。
10. 对于具有子任务的所有题目，将会开启合理的**子任务依赖**。
11. 如果你对于该比赛有任何疑问，请在洛谷上联系负责人 @FFTotoro 或在联考专用 QQ 群（786635884）中进行反馈。

橡子（acorn）

题目描述

龙猫决定在橡树下的小道上收集橡子。

小道可以视为一个数轴。有 n 颗橡子，第 i 颗橡子在数轴上 a_i 的位置上（可能有多个橡子在同一个位置上）。

由于龙猫想要减少工作量，于是她想让你帮她解决如下的问题：

- 定义一次操作为将其中一个橡子在数轴上向左或向右移动一个单位长度。求最少需要多少次操作，使得对于每个橡子，都至少有一个其他的橡子与它位于相同的位置。

形式化地说，每次你可以选择一个 a_i，对其进行操作 $a_i \leftarrow a_i + 1$ 或 $a_i \leftarrow a_i - 1$，求最少的操作次数，使得对于所有 i 存在一个 $j \neq i$ 满足 $a_i = a_j$。

输入格式

从文件 $acorn.in$ 中读入数据。

输入的第一行包含一个整数 n。

输入的第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$。

输出格式

输出到文件 $acorn.out$ 中。

输出一行包含一个整数，表示最少操作次数。

样例 1

【样例输入 1】

```
4
1 2 3 4
```

【样例输出 1】

```
2
```

样例 2

【样例输入 2】

```
5
1 9 8 2 7
```

【样例输出 2】

```
3
```

样例 3

【样例输入 3】

```
10
9 20 5 10 8 1 10 19 15 4
```

【样例输出 3】

```
10
```

样例 4

见选手目录下的 `acorn/acorn4.in` 与 `acorn/acorn4.out`。

该样例满足子任务 3 的性质。

提示

【样例解释 1】

将第 1 个橡子（位置为 1）移动到位置 2，将第 4 个橡子（位置为 4）移动到位置 3。总操作次数为 2，可以证明是最小的。

【样例解释 2】

将第 1 个橡子（位置为 1）移动到位置 2，将第 5 个橡子（位置为 7）移动到位置 8，将第 2 个橡子（位置为 9）移动到位置 8。总操作次数为 3，可以证明是最小的。

【数据范围】

对于所有数据，$2 \leq n \leq 10^5$，$1 \leq a_i \leq 10^9$。

子任务编号	特殊性质	分值
1	$n \leq 4$	17
2	$a_i = i$	20
3	$n \leq 100$，$a_i \leq 100$	35
4	无	28

气球（balloon）

题目背景

龙猫、Jerry 和 V 赖摇斑炸鱼佬准备参加 CCPC。

CCPC 中不可或缺的一环是**气球**；于是 Jerry 提出了一个与气球有关的问题。

题目描述

比赛场馆里设置了 10^9 个位置用来放置气球。初始时，每个位置都是空的。

接下来有 n 个气球批发商前来提供气球：第 i 个批发商将会提供 k_i 个气球，且主办方需要选择 k_i 个在 $[l_i, r_i]$ 间的位置，接着将这 k_i 个气球依次放在这些位置上。

每个位置只能放置**最多一个**气球；确定是否存在一种为每个批发商选择位置的方案，使得最终气球的放置满足上述条件。

输入格式

从文件 $balloon.in$ 中读入数据。

本题包含多组测试数据。

输入的第一行包含一个整数 T，表示测试数据组数。

对于每组数据，输入格式如下：

第一行包含一个整数 n。

接下来 n 行，每行包含三个整数 l_i, r_i, k_i。

输出格式

输出到文件 $balloon.out$ 中。

对于每组数据：输出一行一个字符串 `Yes` 或 `No`，表示是否存在一种满足条件的方案。

样例 1

【样例输入 1】

```
5
3
4 52 40
19 55 2
9 70 29
1
40 68 19
4
41 89 11
5 81 29
5 11 2
70 88 16
5
62 64 1
```

```
64 95 25
69 79 7
35 85 45
9 87 63
2
4 5 1
25 26 1
```

【样例输出 1】

```
No
Yes
Yes
No
Yes
```

样例 2

【样例输入 2】

```
2
3
1 2 1
2 3 1
3 3 1
5
1 2 1
2 3 1
3 3 1
1 3 1
2318 2341 1
```

【样例输出 2】

```
Yes
No
```

样例 3

见选手目录下的 `balloon/balloon3.in` 与 `balloon/balloon3.out`。

该样例满足子任务 $2, 4$ 的限制。

样例 4

见选手目录下的 `balloon/balloon4.in` 与 `balloon/balloon4.out`。

该样例满足子任务 $3, 4$ 的限制。

样例 5

见选手目录下的 `balloon/balloon5.in` 与 `balloon/balloon5.out`。

该样例满足子任务 4 的限制。

提示

【数据范围】

对于所有数据，$1 \le T \le 2 \times 10^5$，$1 \le n \le 2 \times 10^5$，$\sum n \le 6 \times 10^5$，$1 \le l_i \le r_i \le 10^9$，$1 \le k_i \le r_i - l_i + 1$。其中 $\sum n$ 表示 T 组测试数据中 n 的和。

子任务编号	$\sum n \le$	特殊性质	分值
1	6	A	10
2	6×10^5	B	10
3	6×10^5	C	35
4	6×10^3	无	15
5	6×10^5	无	30

- 特殊性质 A：保证 $r_i - l_i + 1 \le 5$ 且 $r_i \le 20$；
- 特殊性质 B：保证 $k_i = r_i - l_i + 1$；
- 特殊性质 C：保证 $k_i = 1$。

蛋糕（cake）

题目描述

龙猫使用收集来的橡子制作了一个圆柱形的橡子蛋糕，于是她决定借机举办一个派对。

派对共有 k 人参加，编号为 1 至 k；这 k 人将要共享该蛋糕。为了将这个蛋糕分给 k 人，龙猫需要沿着蛋糕的底面半径切割 k 次，将蛋糕分为 k 个大小相等的**部分**。

龙猫制作蛋糕时，蛋糕上面已经**均匀**地刻有 $k \times l$ 个切口；她最终只能在这些切口上面切割。切口按照顺时针方向编号为 1 至 kl：我们称切口 $i(1 \leq i < kl)$ 与切口 $(i+1)$ 之间的蛋糕为第 i **小块**，而切口 kl 与切口 1 之间的蛋糕为第 kl 小块。

于是龙猫分蛋糕的过程可以表示为：

- 选择 k 个切口切割蛋糕，将其分为 k 个**均包含 l 个小块**的**部分**，称这个过程为**切割蛋糕**；
- 对于每个人分配恰好一个**部分**，称这个过程为**分配蛋糕**。

由于每个**小块**的口味是不同的，所以接下来这 k 个人将会提出 q 次请求。每次请求的内容如下：

- 对于第 i 次请求，给定两个参数 x_i, y_i，表示编号为 y_i 的人希望在切割蛋糕之后，他 / 她能够分配到包含第 x_i **小块**的**部分**。

龙猫希望计算，在每次请求过后，有多少种分蛋糕的方案：具体地，两种分蛋糕的方案不同，当且仅当在**切割蛋糕**过程中选择的切口构成的集合不同，或者**分配蛋糕**过程中存在某个人被分配到的部分不同。

注意，请求**并不是**独立的，即第 i 次请求过后，你要求出满足前 i 次请求的分蛋糕方案数。保证在所有请求中，x_i 互不相同。

答案对给定的质数 p 取模。

输入格式

从文件 $cake.in$ 中读入数据。

输入的第一行包含两个整数 k, l。

输入的第二行包含一个整数 p。

输入的第三行包含一个整数 q。

接下来 q 行，每行包含两个整数 x_i, y_i。

输出格式

输出到文件 $cake.out$ 中。

输出 q 行，每行包含一个整数，表示每个请求之后的答案对 p 取模的结果。

样例 1

【样例输入 1】

```
3 2
998244353
2
1 2
```

```
6 2
```

【样例输出 1】

```
4
2
```

样例 2

【样例输入 2】

```
8 10
304623133
10
8 8
6 8
51 1
36 5
10 7
38 5
68 3
57 4
76 3
19 2
```

【样例输出 2】

```
50400
40320
5760
960
48
48
12
0
0
0
```

样例 3

【样例输入 3】

```
10 8
446958661
10
26 5
49 9
37 6
10 1
15 3
29 5
69 2
```

```
2 1
25 5
12 1
```

【样例输出 3】

```
2903040
322560
40320
5760
600
240
48
0
0
0
```

样例 4

见选手目录下的 `cake/cake4.in` 与 `cake/cake4.out`。

该样例满足测试点 $5 \sim 6$ 的限制。

样例 5

见选手目录下的 `cake/cake5.in` 与 `cake/cake5.out`。

该样例满足测试点 $7 \sim 10$ 的限制。

样例 6

见选手目录下的 `cake/cake6.in` 与 `cake/cake6.out`。

该样例满足测试点 $11 \sim 15$ 的限制。

提示

【样例解释 1】

在第 1 个请求发出后，有 4 种满足条件的分配方案。以下是一种可能的方案：

- 沿着切口 $1, 3, 5$ 切割蛋糕；
- 将包含 $3, 4$ 的小块分给编号为 1 的人；
- 将包含 $1, 2$ 的小块分给编号为 2 的人；
- 将包含 $5, 6$ 的小块分给编号为 3 的人；

在第 2 个请求发出后，有 2 种满足条件的分配方案。以下是一种可能的方案：

- 沿着切口 $2, 4, 6$ 切割蛋糕；
- 将包含 $4, 5$ 的小块分给编号为 1 的人；
- 将包含 $6, 1$ 的小块分给编号为 2 的人；
- 将包含 $2, 3$ 的小块分给编号为 3 的人。

【数据范围】

对于所有数据，$2 \leq k \leq 2 \times 10^5$，$1 \leq l \leq 2 \times 10^5$，$10^8 < p < 10^9$ 且 p 是质数，
$1 \leq q \leq 2 \times 10^5$，$1 \leq x_i \leq kl$ 且 $x_i \neq x_j (i \neq j)$，$1 \leq y_i \leq k$。

测试点编号	$k \leq$	$l \leq$	$q \leq$
$1 \sim 2$	8	10	10
$3 \sim 4$	100	100	100
$5 \sim 6$	400	400	400
$7 \sim 10$	2500	2500	2500
$11 \sim 15$	10	2×10^5	2×10^5
$16 \sim 20$	2×10^5	2×10^5	2×10^5

基因（dna）

题目描述

Jerry 决定研究龙猫的基因；龙猫的基因序列可以表示为一个长度为 10^9 的整数序列 a。

Jerry 购买了一台机器，用来检测龙猫的基因。具体地，机器可以给出若干条关于基因的信息，每条信息都是如下的形式：

- (l_1, r_1, l_2, r_2) 满足 $1 \le l_1 \le r_1 \le 10^9$ 和 $1 \le l_2 \le r_2 \le 10^9$：表示龙猫的基因满足 $a_{l_1} = a_{l_1+1} = \cdots = a_{r_1-1} = a_{r_1} = a_{l_2} = a_{l_2+1} = \cdots = a_{r_2-1} = a_{r_2}$，即 $a_{l_1 \cdots r_1}$ 与 $a_{l_2 \cdots r_2}$ 中的**所有**元素**两两相等**。

Jerry 想让你从已有的信息中推断出一些其他的信息。具体地，使用机器的过程中**依次**发生了 n 次事件，每次事件可能是以下的两种之一：

- Type 1：机器给出了一条形如 (l_1, r_1, l_2, r_2) 的信息；
- Type 2：Jerry 提出一组询问 (x, y)，根据目前机器给出的所有信息，是否能**确定** $a_x = a_y$。

请你帮他处理这些事件。

输入格式

从文件 $dna.in$ 中读入数据。

输入的第一行包含一个整数 n。

接下来输入 n 行，每行均为如下的形式之一：

- $1\ l_1\ r_1\ l_2\ r_2$：表示 Type 1 事件；
- $2\ x\ y$：表示 Type 2 事件。

输出格式

输出到文件 $dna.out$ 中。

对于每个 Type 2 事件，输出一行一个字符串 `Yes` 或 `No`，表示是否能确定 $a_x = a_y$。

样例 1

【样例输入 1】

```
11
1 2 3 7 8
2 2 3
2 2 8
2 1 4
1 3 4 6 6
2 4 9
2 4 8
1 1 1 9 10
2 1 2
1 8 9 8 8
2 1 2
```

【样例输出 1】

```
Yes
Yes
No
No
Yes
No
Yes
```

样例 2

见选手目录下的 `dna/dna2.in` 与 `dna/dna2.out`。

该样例满足子任务 2 的限制。

样例 3

见选手目录下的 `dna/dna3.in` 与 `dna/dna3.out`。

该样例满足子任务 4, 5 的限制。

样例 4

见选手目录下的 `dna/dna4.in` 与 `dna/dna4.out`。

该样例满足子任务 5 的限制。

提示

【数据范围】

对于所有数据，$1 \leq n \leq 10^6$，$1 \leq l_1 \leq r_1 \leq 10^9$，$1 \leq l_2 \leq r_2 \leq 10^9$，$1 \leq x, y \leq 10^9$。保证至少有一个 Type 2 事件。

子任务编号	$n \leq$	特殊性质	分值
1	10^6	A	2
2	300	B	5
3	3000	B	8
4	10^6	C	15
5	3000	无	10
6	6×10^4	无	20
7	1.5×10^5	无	20
8	10^6	无	20

- 特殊性质 A：保证没有 Type 1 事件；
- 特殊性质 B：保证输入的所有数不超过 n；

- 特殊性质 C：保证对于所有 Type 1 事件，$l_1 = l_2$ 且 $r_1 = r_2$。