

NOIP模拟赛

题目名称	猫捉老鼠	格斗大赛	大脑过载	龙了个龙
题目类型	传统型	传统型	传统型	传统型
目录	catch	botbox	brain	dragon
可执行文件名	catch	botbox	brain	dragon
输入文件名	catch.in	botbox.in	brain.in	dragon.in
输出文件名	catch.out	botbox.out	brain.out	dragon.in
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	256 MB	256 MB	256 MB	256 MB
测试点数目	10	20	25	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	catch.cpp	botbox.cpp	brain.cpp	dragon.cpp
-----------	-----------	------------	-----------	------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项与提醒（请选手务必仔细阅读）

1. 选手提交的源程序必须存放在**已建立好的，且带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**
5. 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**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

猫捉老鼠 (catch)

【题目描述】

屋里出现了一只老鼠，困扰了小 x 一家。为了消灭这只老鼠，他们决定采用生物方法。为找来老鼠的天敌，他们痛下血本，买了一只家猫。据卖家说，这只猫还是当年的捕鼠冠军汤姆猫的后代，捕鼠能力强不强不知道，但一定会用你想不到的方法捕鼠。

这只非同寻常的猫花了几天时间，在屋外布置好了一个迷宫。这个迷宫共有 n 个区域，从 1 到 n 进行编号。其中还有 n 条双向通道，第 i 条通道连接两个区域 $u_i, v_i (u_i < v_i)$ 。不存在两条通道连接的区域相同。在迷宫中只能通过这些通道来移动，一次移动可以从某条通道连接的其中一个区域到达它连接的另一个区域。从任意一个区域出发都能进行若干次移动到达任意另外一个区域。

猫运用起了祖传的“美食计”，将会把老鼠从屋内老巢引诱到迷宫中编号为 m 的区域内，并会触发这个区域中的神秘机关，使老鼠无法逃出迷宫。而猫自己潜伏在编号为 c 的区域内，暗中观察，静候良机。为了不打草惊蛇，机关所在的区域和猫埋伏的区域不是同一个区域。

老鼠果然中计，但也以惊人的反应力和大脑转速弄清了迷宫构造。猫也已经发现目标。一场猫鼠追逐战即将打响。

猫鼠的一次行动会进行一次移动或者不动。一开始，老鼠会行动一次，接着猫行动一次，然后老鼠再行动一次，接下来猫又行动一次……以此类推，老鼠和猫交替行动。

迷宫不大，也就最多 10^6 个区域，猫鼠随时可以看到对方在哪个位置。老鼠会尽可能地逃命，猫会尽可能地捉到老鼠。当老鼠和猫在同一个区域的时候，猫就捉到老鼠了。

屋里的小 x 看着屋外的情景惊呆了，他想知道这只猫能不能在有限行动次数内捉到老鼠。请你告诉他。

就算猫不能捉到老鼠，这只被困在迷宫内的老鼠也会被小 x 正义制裁。奇怪的是，这只老鼠被消灭后不久，屋里又出现了新的一只老鼠。虽然迷宫可以用之前的，但是机关是一次性的，所以猫会重新在一个区域设置机关，在另一个区域埋伏，不一定和之前的是同一个区域。消灭新老鼠后，竟发现还有一只……在前前后后出现了 Q 只老鼠后，小 x 一家终于无法忍受每天的猫鼠大战，搬家了。

那么请你回答 Q 个问题。

简要题意：给定一个 n 个点 n 条边的无向连通图，进行 Q 次询问。每次询问初始时，图中 c 点上有一只猫， m 点上有一只老鼠。从老鼠开始，老鼠和猫交替行动，每次行动移动到所在点的相邻一个点或者不移动。猫和老鼠在同一个点时，猫会捉到老鼠。问猫能否在有限行动次数内保证捉到老鼠。

【输入格式】

从文件 *catch.in* 中读入数据。

第一行包含两个整数 n, Q ，表示迷宫区域及通道的数量和老鼠数量。

接下来 n 行，第 i 行包含两个整数 u_i, v_i ，表示第 i 个通道连接的两个区域编号。

接下来 Q 行，第 i 行包含两个整数 m_i, c_i ，表示猫捉第 i 只老鼠时，机关所在位置和潜伏的位置。

【输出格式】

输出到文件 *catch.out* 中。
共 Q 行。若猫能在有限行动次数内捉到第 i 只老鼠，在第 i 行输出 ‘Yes’，否则在第 i 行输出 ‘No’。

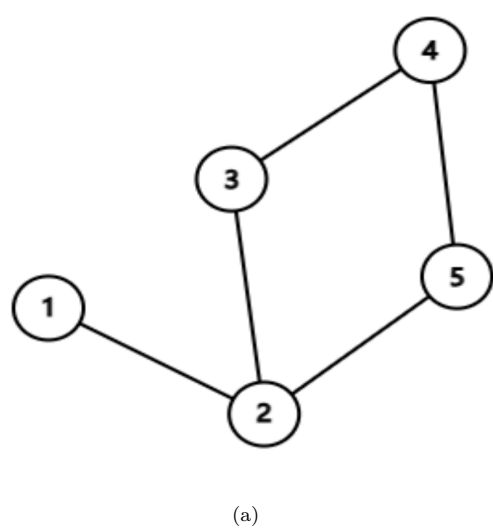
【样例 1 输入】

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

【样例 1 输出】

```
1 Yes
2 No
```

【样例 1 解释】



迷宫的结构如图所示。

第 1 只老鼠一开始在 1 号区域，猫一开始在 3 号区域。若老鼠移动到 2 号区域，猫可以直接移动到 2 号区域捉到老鼠；若老鼠不移动，猫可以移动到 2 号区域，接下来老鼠要想活命就需要不移动，猫就移动到 1 号区域捉到老鼠。所以无论如何猫能在有限行动次数内捉到老鼠。

第 2 只老鼠一开始在 1 号区域，猫一开始在 4 号区域。老鼠移动到 2 号区域，此后只需要在 2, 3, 4, 5 四个区域中活动，且保证在猫行动前老鼠所在区域不与猫所在区域有通道相连，就能让猫捉不到老鼠。

【样例 2 输入】

见附加文件里的 catch/catch.in

【样例 2 输出】

见附加文件里的 catch/catch.out

【数据范围】

对于所有数据，保证： $3 \leq n \leq 10^6$, $1 \leq Q \leq 10^6$ ； $1 \leq u_i < v_i \leq n$ ，任意两条通道 $i, j (i \neq j)$ ，满足 $u_i \neq u_j$ 或 $v_i \neq v_j$ ； $1 \leq m_i, c_i \leq n$ ， $m_i \neq c_i$ 。

测试点编号	$n \leq$	$Q \leq$
1, 2	10	10
3, 4	1000	1000
5, 6	5×10^4	5×10^4
7, 8	10^5	10^5
9, 10	10^6	10^6

格斗大赛 (botbox)

【题目描述】

第 42 届“金氪拉”杯机器人格斗大赛开幕了。

这场盛大的赛事共有 n 支队伍参赛，这是前所未有的规模。但这场比赛的奇特之处不仅在于规模的盛大，还在于他奇怪到有些匪夷所思的规则：

比赛开始时，编号为一的队伍作为守擂者，其他的参赛队伍按编号从小到大依次进入等待队列。接下来其他队伍将向守擂者发起 k 次挑战。每次挑战中守擂者将对战等待队列中最靠前的一支队伍，失败的队伍将回到等待队列的末尾，而胜利的队伍将成为新的守擂者。（因为守擂者有主场优势，所以就算平局，也视作守擂者胜利）在 k 次挑战后最终的守擂者将成为整场比赛的胜利者。

小 z 观察后发现，参加比赛的队伍使用的机器人大致可以分为三种类型，不妨把他们叫做 A 类机器人、B 类机器人和 C 类机器人。不同类型的机器人有不同的特点，从而导致了不同类型的机器人之间有着一定的克制关系。由于比赛中机器人的强度都不相上下，机器人之间的克制关系似乎成为了决定胜负的唯一因素。因此，A 类机器人一定能打败 B 类机器人，B 类机器人一定能打败 C 类机器人，C 类机器人一定能打败 A 类机器人，相同类型的机器人一定会达成平局。

在发现了规律后，小 z 顿时失去了观看这场比赛的欲望，因为根据现有的信息（每支队伍使用的机器人的类型），已经可以推断出胜利者是谁了。同时，他也不关心每支队伍的战况。但他还是很好奇在比赛结束后，胜利的队伍和等待队列中从前往后每一支队伍使用的机器人的类型分别是什么。

【输入格式】

从文件 `botbox.in` 中读入数据。

第一行两个整数 n 和 k ，表示参赛队伍的数量。

第二行一个字符串 s 。 s_i 表示编号为 i 的队伍使用的机器人的类型。

【输出格式】

输出到文件 `botbox.out` 中。

一行一个字符串 t 。 t_1 表示比赛结束后胜利的队伍使用的机器人的类型。 t_{i+1} ($1 \leq i < n$) 表示比赛结束后等待序列中从前往后第 i 支队伍使用的机器人的类型。

【样例 1 输入】

```
1 5 3
2 ACABC
```

【样例 1 输出】

```
1 BCAAC
```

【样例 1 解释】

第一次挑战中，守擂者是编号为 1 的队伍，使用的是 A 型机器人。功擂者是编号为 2 的队伍，使用的是 C 型机器人。编号为 2 的队伍胜利，成为新的守擂者，编号为 1 的队伍到等待队列的末尾，此时等待队列中的队伍的编号从前往后依次是 3, 4, 5, 1。

第二次挑战中，守擂者是编号为 2 的队伍，使用的是 C 型机器人。功擂者是编号为 3 的队伍，使用的是 A 型机器人。守擂者胜利。

第一次挑战中，守擂者是编号为 2 的队伍，使用的是 C 型机器人。功擂者是编号为 4 的队伍，使用的是 B 型机器人。功擂者胜利。

最终的胜利者是编号为 4 的队伍，等待队列中的队伍的编号从前往后依次是 5, 1, 3, 2。

【样例 2 输入】

```
1 10 15
2 BCBABCCACA
```

【样例 2 输出】

```
1 CACACCBBA
```

【样例 3 输入】

```
1 3 100
2 ABC
```

【样例 3 输出】

```
1 ACB
```

【数据范围】

对于所有数据，保证 $2 \leq n \leq 10^6$ ， $k \leq 10^{18}$ ， s 中只包含 ‘A’、‘B’、‘C’ 三种字符。

测试点编号	$n \leq$	$k \leq$	特殊性质
1	10	10	无
2 ~ 3	200	200	无
4 ~ 6	10^6	10^7	无
7 ~ 8	10^6	10^9	无
9 ~ 10	200	10^{18}	无
11 ~ 13	10^6	10^{18}	$k = 10^{18}$
14 ~ 17	10^6	10^{18}	$\sum_{i=1}^{n-1} [s_i \neq s_{i+1}] \leq 100$
18 ~ 20	10^6	10^{18}	无

大脑过载 (brain)

【题目描述】

可怜的小 z 被外星人抓走了。

外星人认为小 z 看起来很好吃，但是星际条约规定不能伤害其他拥有智能的物种。于是为了检测智力，外星人给了小 z 这样一个问题：

给你一个有 n 个节点的有根树，树根编号 1。编号为 i 的节点点权是 a_i ，他的父亲的编号为 fa_i ，他到他的父亲的边的边权为 w_i 。

定义 $f(x, y)$ 为结点 x 到结点 y 的路径上的边权和，一棵树的值为 $\sum_{i=1}^{n-1} \sum_{j=i+1}^n f(i, j) a_i a_j$ 。

定义对结点 i 的改造操作如下：在以 i 为根的子树中选择一个结点 x ，断开边 (i, fa_i, w_i) ，再连接边 (x, fa_i, w_i) ，得到一个新的有根树。

对于每个非树根结点 i ，你要求出对结点 i 进行一次改造操作后，得到的树的价值的最大值。询问之间独立。

小 z 没见过这种大场面，被吓得大脑宕机，所以只能靠你来解决这个问题了。不然的话，下一个被抓走的可能就是你。

【输入格式】

从文件 `brain.in` 中读入数据。

第一行一个整数 n ，表示结点数量。

第二行 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ，表示每个点的点权。

接下来 $n-1$ 行，其中第 i 行包含两个整数 fa_{i+1} 和 w_{i+1} ，分别表示编号为 $i+1$ 的点的父亲编号、编号为 $i+1$ 的节点到 fa_{i+1} 的边的边权。

【输出格式】

输出到文件 `brain.out` 中。

输出共 $n-1$ 行，第 i 行包含一个整数，表示对结点 i 进行一次改造操作后得到的树的价值的最大值。

【样例 1 输入】

```
1 4
2 2 3 1 4
3 1 1
4 2 3
5 2 2
```

【样例 1 输出】

```
1 127
2 91
```

3

91

【样例 1 解释】

如果对节点 2 进行改造操作，选择 $x = 3$ 可以得到最大价值。

【样例 2 输入】

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

【样例 2 输出】

```
1 63
2 33
3 69
4 33
5 33
```

【样例 2 解释】

如果对节点 2 进行改造操作，选择 $x = 6$ 可以得到最大价值。

如果对节点 4 进行改造操作，选择 $x = 6$ 可以得到最大价值。

【数据范围】

对于全部数据，保证 $1 \leq n \leq 10^5$ ， $1 \leq fa_i < i$ ， $-10^4 \leq a_i, w_i \leq 10^4$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 4	100	无
5 ~ 8	2000	无
9 ~ 17	10^5	$fa_i = i - 1$
18 ~ 21	10^5	$a_i, w_i \geq 0$
22 ~ 25	10^5	无

龙了个龙 (dragon)

【题目描述】

舞龙高手铁掌大师在为熊大熊二介绍龙时说：“龙！这是龙！中国龙！它们代表了中国神话中的神兽！你们这两个丛林土鳖！”“龙不是动物，而是一种精神。”龙文化离我们不远，最近小 x 就迷上了一款名为“龙了个龙”的游戏。

“龙了个龙”有多种游戏模式，玩法多样。

“经典模式”的玩法如下：游戏开始时，系统会给你 n 颗排成一行的宝石，从左到右第 i 颗宝石的颜色为 a_i ，并给定一个参数 k ，保证 n 是 k 的倍数。玩家每次使用 k 颗颜色相同的宝石合成一条龙，每颗宝石在一局“经典模式”游戏中只能使用一次。玩家用完所有宝石则游戏胜利。这游戏怎么输啊？“龙了个龙”作为老少皆宜的休闲游戏，任何模式的失败条件都是玩家主动放弃。

恶心的是，很多时候，系统给的宝石颜色让这个游戏根本无法胜利，被玩家们称为“无解关”。当然，不是“无解关”的关卡被称为“有解关”。小 x 深受其害，他曾亲眼看见系统给了 3 颗宝石，每颗宝石的颜色分别为 1, 1, 4，参数 k 为 3……

游戏因此受到了差评，于是游戏官方为改善玩家游戏体验，推出了“挑战模式”，玩法如下：游戏开始时仍然是 n 颗有颜色的宝石及参数 k ，这里就不保证 n 是 k 的倍数了。玩家需要自选一个区间 $[l, r]$ ，应保证区间长度，即 $r - l + 1$ ，是 k 的倍数，然后用区间内的这些宝石和参数 k 进行一次“经典模式”的游戏。玩家多次选择区间，多次进行“经典模式”游戏，但需要保证每次“经典模式”游戏选择的区间互不相同。玩家需要找出所有的“有解关”区间进行“经典模式”游戏，所有“有解关”的游戏都胜利后，玩家胜利。

但是小 x 更喜欢玩的是“无尽模式”，玩法如下：游戏开始时还是 n 颗有颜色的宝石和参数 k ， n 也不一定是 k 的倍数。“无尽模式”自然有无限轮，每一轮游戏系统会选定一个区间 $[l, r]$ ，玩家需要用这个区间内的宝石和参数 k 进行一次“挑战模式”的游戏。该“挑战模式”游戏胜利后，“无尽模式”游戏进入下一轮。据说目前“无尽模式”的最高纪录是 1437 轮。

孤军奋战多没意思，小 x 告诉你“无尽模式”中系统给的宝石数量 n 、每颗宝石的颜色 a_i 以及参数 k ，还有每轮游戏系统选定的 $[l, r]$ 。你需要告诉他，每轮“无尽模式”游戏中，小 x 需要玩多少次“经典模式”游戏。

小 x 玩了 Q 轮后，感觉游戏枯燥，无挑战性，终于卸载了游戏。但是你还是需要回答这 Q 个问题。

简要题意：给定一个含 n 个元素的序列 a 以及一个参数 k ， Q 次询问，每次询问查询一个区间 $[l, r]$ 中有多少个子区间内所有元素的出现次数均为 k 的倍数。

【输入格式】

从文件 `dragon.in` 中读入数据。

第一行包含三个整数 n, k, Q ，表示宝石数量、游戏中的参数和游戏轮数。

第二行包含 n 个整数，第 i 个整数为 a_i ，表示从左到右第 i 颗宝石的颜色。

接下来 Q 行，每行包含两个整数 l_i, r_i ，表示第 i 轮游戏中，系统选定的区间左右端点。

【输出格式】

输出到文件 *dragon.out* 中。

共 Q 行，第 i 行包含一个整数，表示第 i 轮游戏中小 x 需进行的“经典模式”游戏次数。

【样例 1 输入】

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

【样例 1 输出】

```
1 2
2 3
```

【样例 1 解释】

第 1 轮游戏中，选择区间 $[2, 5]$ 中的宝石进行“挑战模式”游戏，4 颗宝石颜色分别为 2, 1, 1, 2。再从这 4 颗宝石中选择一个长度是 k 的倍数且是“有解关”的区间进行“经典模式”游戏。合法区间有 $[2, 3], [1, 4]$ 。其中 $[2, 3]$ 是 2 颗颜色为 1 的宝石，颜色相同，可以合成一条龙； $[1, 4]$ 是 2 颗颜色为 2 的宝石以及 2 颗颜色为 1 的宝石，颜色为 2 的合成一条龙，颜色为 1 的合成另一条龙。另外两个长度为 k 的倍数的区间 $[1, 2], [3, 4]$ ，可以证明是“无解关”。

第 2 轮游戏中小 x 需在区间 $[1, 6], [2, 5], [3, 4]$ 中进行“经典模式”游戏。

【数据范围】

对于全部数据，保证 $1 \leq k \leq n \leq 10^5, 1 \leq Q \leq 10^5, 1 \leq a_i \leq n, 1 \leq l_i \leq r_i \leq n$ 。

测试点编号	$n \leq$	$Q \leq$	特殊性质
1 ~ 4	100	100	无
5 ~ 6	500	500	无
7 ~ 8	1000	1000	无
9 ~ 13	5×10^4	5×10^4	无
14 ~ 16	10^5	1	$l_i = 1, r_i = n$
17	10^5	10^5	$k = 1$
18 ~ 20	10^5	10^5	$k = 2$
21 ~ 25	10^5	10^5	无