NOIP2023 模拟赛

题目信息

题目名称	数字游戏	过河卒II	树图	异或区间
题目类型	传统题	传统题	传统题	传统题
可执行文件名	game	pawn	diagrams	chemistry
输入文件名	game.in	pawn.in	diagrams.in	chemistry.in
输出文件名	game.out	pawn.out	diagrams.out	chemistry.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	20	7	3	20
子任务是否等分	是	否	否	是
是否开启捆绑测试	否	是	是	否

编译选项

对于 C++ 语言，`-std=c++14 -O2`。

注意事项

1. 选手提交的源程序请直接放在个人目录下，无需建立子文件夹。
2. 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。
3. 程序可使用的栈空间大小与该题内存空间限制一致。
4. 若无特殊说明，每道题的代码大小限制为 100 KB。
5. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
6. 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避免出错。
7. 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。
8. 只提供 Linux 格式附加文件，同时评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

数字游戏

题目描述

给定 n 个正整数 a_i。

接下来有 q 次询问，第 j 次询问给出正整数 p_j。

每次询问时有一个可重集 S，S 初始为 a 的前 p_j 个数，还有一个变量 c_j 初始为 0。

然后进行 n 次操作，第 k 次操作时，令 x 为 S 的最大值，c_j 加上 $(-1)^{k-1}x$，然后将 x 从 S 中删除，最后如果 $p_j+k\le n$，则在 S 中插入数字 a_{p_j+k}。

注意 S 中若有多个相同的 x，那么只删除其中一个。

操作结束后回答 c_j 的值。

输入格式

第一行输入两个正整数 n,q。

第二行输入 n 个正整数 a_i。

第三行输入 q 个正整数 p_j。

输出格式

输出 q 行，每行一个整数 c_j。

样例 1

输入

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

输出

```
6
```

解释

对于第 1 个询问：

S 初始为 $\{2,4,2\}$，c_1 初始为 0。

第 1 次操作，$x=4$，c_1 加上 4，c_1 变成 4，S 删除 4，S 插入 3，S 变成 $\{2,2,3\}$。

第 2 次操作，$x=3$，c_1 加上 -3，c_1 变成 1，S 删除 3，S 插入 5，S 变成 $\{2,2,5\}$。

第 3 次操作，$x=5$，c_1 加上 5，c_1 变成 6，S 删除 5，S 不插入数字，S 变成 $\{2,2\}$。

第 4 次操作，$x=2$，c_1 加上 -2，c_1 变成 4，S 删除 2，S 不插入数字，S 变成 $\{2\}$。

第 5 次操作，$x=2$，c_1 加上 2，c_1 变成 6，S 删除 2，S 不插入数字，S 变成 $\varnothing$。

所以最后输出 6。

数据范围与提示

对于所有数据，$1 \le n \le 10^5, 1 \le q \le 2 \times 10^3, q \le n, 1 \le a_i, p_j \le n$。

- 对于 10% 的数据，$n \le 10$。
- 对于 30% 的数据，$n \le 600$。
- 对于 50% 的数据，$n \le 10^4, q \le 10^3$。

过河卒II

题目描述

给出一个 n 行 m 列的网格，行编号为 0 到 $n-1$，列编号为 0 到 $m-1$。

记坐标 (x,y) 表示行编号为 x，列编号为 y 的格子，该格子上面写有自然数 $a_{x,y}$。

有 k 个特殊格子，第 i 个特殊格子坐标为 (r_i,c_i)，不存在两个坐标相同的特殊格子。

第 i 个特殊格子有以下四种选择周围格子的方式，每个特殊格子将执行其中一种可行的选择方式：

1. 选择 $(r_i,c_i),(r_i,c_i-1),(r_i+1,c_i),(r_i-1,c_i)$ 这 4 个格子。
2. 选择 $(r_i,c_i),(r_i,c_i+1),(r_i+1,c_i),(r_i-1,c_i)$ 这 4 个格子。
3. 选择 $(r_i,c_i),(r_i,c_i+1),(r_i,c_i-1),(r_i-1,c_i)$ 这 4 个格子。
4. 选择 $(r_i,c_i),(r_i,c_i+1),(r_i,c_i-1),(r_i+1,c_i)$ 这 4 个格子。

一种选择方式可行，当且仅当该方式选择的 4 个格子均在网格内。

特别的，如果一个特殊格子不存在一种可行的选择方式，输出 `No` 即可。

当每个特殊格子都选择一种可行的选择方式后，我们称这是一种方案。

一个方案是可行的，当且仅当每个格子（包括特殊格子）都被至多 1 个特殊格子选择。允许一个格子不被任何特殊格子选择。

定义一个可行的方案的权值是所有被选择的格子（包括特殊格子）的 $a_{x,y}$ 之和。

求出所有可行的方案的权值最大值。

特别的，如果不存在一种可行的方案，输出 `No` 即可。

输入格式

第一行输入两个正整数 n,m。

接下来 n 行，每行读入 m 个自然数，代表格子上写的数字 $a_{x,y}$。

第 $n+2$ 行读入一个正整数 k。

接下来 k 行，每行读入 2 个自然数 r_i,c_i。

输出格式

输出一行一个字符串 `No`，或者输出一行一个整数表示所有可行的方案的权值最大值。

样例 1

输入

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

```
1 1
2 2
```

输出

```
20
```

解释

一种可行的方案是：

第 1 个特殊格子选择 $(1,1),(1,0),(1,2),(0,1)$。

第 2 个特殊格子选择 $(2,2),(2,1),(2,3),(3,2)$。

该方案的权值为 $2+2+3+4+4+1+2+2=20$。

样例 2

输入

```
1 1
0
1
0 0
```

输出

```
No
```

解释

唯一的一个特殊格子 $(0,0)$ 不存在一种可行的选择方式，所以输出 `No`。

样例 3

输入

```
4 4
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
3
1 1
2 2
1 2
```

输出

```
No
```

数据范围与提示

对于 100% 的数据，保证 $1 \le k \le nm \le 10^6, 0 \le a_{x,y} \le 10^3, 0 \le r_i < n, 0 \le c_i < m$，且不存在坐标相同的特殊格子。

子任务编号	k	特殊限制	分值
1	$\le 10^3$	保证 $\forall 1 \le i < j \le k, \max(\lvert r_i - r_j \rvert, \lvert c_i - c_j \rvert) > 2$。	5
2	$\le 10^3$	保证 $\forall 1 \le i < j \le k$，若 $\max(\lvert r_i - r_j \rvert, \lvert c_i - c_j \rvert) \le 2$，则 (r_i, c_i) 和 (r_j, c_j) 是上下或左右相邻的格子。	10
3	$\le 10^3$	保证 $\forall 1 \le i < j \le k, \max(\lvert r_i - r_j \rvert, \lvert c_i - c_j \rvert) \ne 2$。	10
4	$\le 10^3$	保证 $\forall 1 \le i < j \le k, r_i = r_j$。	10
5	≤ 10		15
6	$\le 10^3$		20
7			30

如果你对如何存储网格 a 有疑惑，可以使用：

```
std::vector<std::vector<int>> a(n, std::vector<int>(m));
```

来创建一个 n 行 m 列的初始值为 0 网格 a，行的访问范围是 0 到 $n-1$，列的访问范围是 0 到 $m-1$。

树图

题目描述

有一棵 n 个点的树，每个点有状态 $0/1/2$，所有点初始状态为 0，第 i 条边连接点 u_i 和 v_i。

定义树的边集的一个子集 S 合法，当且仅当在原树删去属于 S 的边形成的图中，不存在一个点 x 的状态为 1，另一个点 y 的状态为 2，且 x 与 y 连通。

定义当前树的权值是：所有合法子集 S 中，S 大小的最小值。

有 q 次修改点的状态，第 j 次修改的类型是 t_j，修改的点是 x_j。总共有三种类型的修改：

1. $t_j = 1$，将点 x_j 的状态修改为 1。保证此时点 x_j 的状态为 0。
2. $t_j = 2$，将点 x_j 的状态修改为 2。保证此时点 x_j 的状态为 0。
3. $t_j = 3$，将点 x_j 的状态修改为 0。保证此时点 x_j 的状态不为 0。

在每次修改后，求出当前树的权值。

输入格式

第一行输入一个正整数 n。

接下来 $n-1$ 行，每行输入两个正整数 u_i, v_i。

第 $n+1$ 行输入一个正整数 q。

接下来 q 行，每行输入两个正整数 t_j, x_j。

输出格式

输出 q 行，每行输出一个整数表示当前树的权值。

样例 1

输入

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

输出

```
0
1
2
3
```

解释

第一个询问选择 $S = \varnothing$ 即可，输出 0。

第二个询问选择 $S = \{(1, 2)\}$ 即可，输出 1。

第三个询问选择 $S = \{(1, 2), (3, 4)\}$ 即可，输出 2。

第四个询问选择 $S = \{(1, 2), (3, 4), (3, 5)\}$ 即可，输出 3。

数据范围与提示

对于 100% 的数据，保证 $1 \leq n, q \leq 100000, 1 \leq u_i, v_i, x_j \leq n, t_j \in \{1, 2, 3\}$。

子任务编号	分值	n	q
1	10	≤ 15	≤ 100
2	30	≤ 1000	≤ 1000
3	60	≤ 100000	≤ 100000

异或区间

题目描述

给出 n 个不交区间 $[l_i, r_i]$，令 $S = \bigcup\limits_{i=1}^{n} ([l_i, r_i] \cap \mathbb{Z})$。

给出正整数 k，求出：

$$\left(\sum_{a \in S} \sum_{b \in S} \sum_{c \in S} [a < b < c][(a \oplus b) \le k][(b \oplus c) \le k][(c \oplus a) \le k] \right) \pmod{(10^9 + 7)}$$

其中 $[\text{condition}]$ 在条件 condition 为真时为 1，否则为 0。

输入格式

第一行输入两个正整数 n, k。

接下来输入 n 行，每行输入两个自然数 l_i, r_i。

输出格式

输出一行一个整数表示答案。

样例 1

输入

```
2 5
1 5
6 10
```

输出

```
11
```

解释

11 个可能的 a, b, c 分别是：
$(1, 2, 3), (1, 4, 5), (2, 3, 6), (2, 3, 7), (2, 6, 7), (3, 6, 7), (4, 5, 6), (4, 5, 7), (4, 6, 7), (5, 6, 7), (8, 9, 10)$
。

数据范围与提示

对于 100% 的数据，保证
$1 \le n \le 20000, 0 \le l_i \le r_i \le 10^9, 1 \le k \le 10^9, \forall 1 \le i < n, r_i < l_{i+1}$。

测试点编号	特殊限制
1	$\max(k, r_n) \le 500$
$2, 3$	$\max(k, r_n) \le 10^4$

测试点编号	特殊限制
$4, 5$	存在正整数 x，使 $k = 2^x - 1$。
$6, 7, 8, 9, 10$	$\max(k, r_n) \leq 10^6$
$11, 12, 13, 14, 15, 16$	$n \leq 20$
$17, 18, 19, 20$	