题目名称	Qrj 的排列	Mt_F 的矩阵	Fun_S 的有向图	Sixy 的神秘机器
源程序文件名	permutation.cpp	matrix.cpp	path.cpp	machine.cpp
输入文件名	permutation.in	matrix.in	path.in	machine.in
输出文件名	permutation.out	matrix.out	path.out	machine.out
时间限制	1s	1s	1s	3s
内存限制	512MB	512MB	512MB	1024MB
是否捆绑测试	是	否	否	否
是否有SPJ	否	否	否	否
题目类型	传统型	传统型	传统型	传统型

编译选项：`-lm -std=c++14 -O2 -Wl,--stack=2147483647`

出题人提示：不要尝试搜索题目中出现的名字，你会大败而归。

A：Qrj 的排列

题目描述

Qrj 有一个长为 n 的排列 a。

定义一个长为 n 的数列为一个排列，当且仅当这个数列中 1 到 n 每个数都出现过一次。

现在他可以对这个排列做出多次如下操作：

选定排列中连续的三个位置，并将其翻转，例如排列 `1,2,3,4,5`，可以选中第二到四个位置翻转为 `1,4,3,2,5`。

现在给定这个排列，需要通过数次操作将其还原为从 1 到 n，试问最少需要进行多少次操作能还原，或者根本不能还原，如果不能还原则输出 -1。

输入格式

第一行一个正整数 n 表示排列长度。

接下来一行 n 个正整数 a_i 表示这个排列。

输入格式

输出一个整数，表示最小操作次数，如果不能完成输出 -1。

样例

样例输入 #1

```
4
3 2 1 4
```

样例输出 #1

```
1
```

样例输入 #2

```
3
1 3 2
```

样例输出 #2

```
-1
```

数据范围

本题开启捆绑测试。

子任务	n	特殊性质	分值
1	$\le 5\times 10^5$	$\forall 1\le i\le n, a_i=i$	1
2	≤ 4	保证排列是出题人手搓的	4
3	≤ 20	保证排列随机生成	10
4	$\le 5\times 10^5$	$\forall 1\le i\le n, a_i=n+1-i$	20
5	$\le 5\times 10^5$	$a_1-a_2=2$	5
6	≤ 2000	保证排列在有解情况下随机生成	20
7	$\le 5\times 10^5$	无	40

对于所有数据，有 $3\le n\le 5\times 10^5, 1\le a_i,b_i,c_i\le 3, 1\le p_i\le 10^9$ 。保证 a 是一个排列。

B：Mt_F 的矩阵

题目描述

Mt_F 有一个 $n\times m$ 的矩阵，它是有初始值的。

同时还有 q 次询问，每次询问矩阵的某一个子矩阵内部所有元素的 $\gcd$。

由于 Mt_F 太唐氏了，所以没有修改操作。

输入格式

第一行三个数 n、m、q，分别表示矩阵的行数，列数和询问的个数。

接下来 n 行，每行 m 个正整数，表示矩阵的每个数。

接下来 q 行，每行四个数 $x_1 \quad y_1 \quad x_2 \quad y_2$，表示询问以 (x_1, y_1) 为左上角，以 (x_2, y_2) 为右下角的矩阵中所有元素的 $\gcd$。

输出格式

输出 q 行，每行一个数。表示每次询问的答案。

样例 #1

样例输入 #1

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

样例输出 #1

```
1
1
3
```

温馨提示

数据掺杂了一些水分， 所以提供了一点神奇做法的可能性。

数据范围

本题不开启捆绑测试。

测试点	n, m	q	特殊性质
$1 \sim 2$	≤ 10	≤ 10	无
$3 \sim 4$	≤ 300	$\le 10^5$	无
$5 \sim 10$	≤ 500	$\le 10^5$	无

对于 100% 的数据，保证所有数据在 int 范围以内。

C：Fun_S 的有向图

题目描述

Fun_S 有一个由 n 个点和 m 条边构成的**有向图** G。

对于这个有向图，请你找到一条从 1 到 n 的路径，使得这条路径上第 k 长的边尽可能长。

注意：一条路径允许重复经过某些边。相应地，在计算第 k 大的边时，这些重复经过的边也会被算**多次**。如果一条路径经过的边数不足 k，那么我们认为这条路径第 k 长 的边长度为 0。一条路径允许多次经过 1 和 n，只要起始点为 1，最终落在 n 即可。

输入格式

第一行两个数字 n, m, k，n, m 表示有向图的点数、边数，k 的意义如上所述。

接下来 m 行，每行三个数 u, v, w，表示有向图中存在一条由 u 指向 v，长度为 w 的边。

输出格式

一个数，表示这条路径第 k 长的边的长度。

如果这样的边不存在，输出 -1。

样例 #1

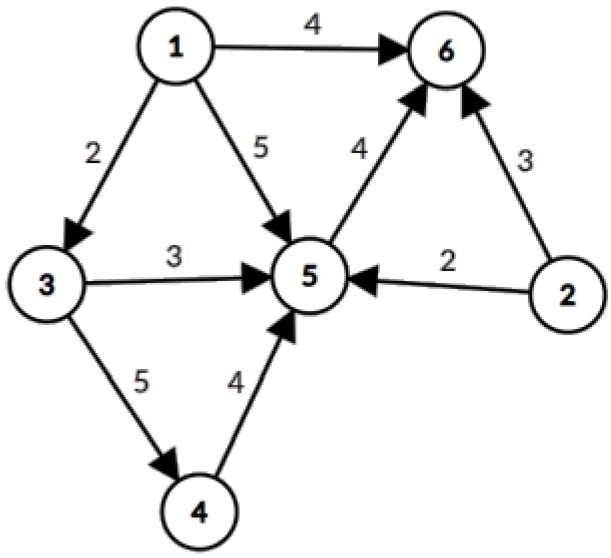
样例输入 #1

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

样例输出 #1

```
4
```

样例 #1 解释



显然，$1 \to 3 \to 4 \to 5 \to 6$ 就是我们要找的那条路径。

数据规模

测试点编号	$n \le$	$m \le$	$k \le$	特殊性质
$1 \sim 4$	6	10	10	无
$5 \sim 6$	1000	2000	1	保证输入为**有向无环图**
$7 \sim 8$	1000	5000	100	保证输入为**有向无环图**
$9 \sim 12$	10^5	5×10^5。	1	无
$13 \sim 20$	10^5	5×10^5。	10^6	无

对于 100%。的数据，保证 $1 \le w \le 10^9$，不保证 1 能到达 n。

D：Sixy 的神秘机器

题目描述

Sixy 进入一个神秘的区域，在这里他发现了一个神秘的机器。

这个机器以一个数组的样子存在，它的长度为 n，第 i 个位置的值为 $machine_i$。

这个机器的原理十分神秘，它可以执行一些操作。对于一次操作 `operate(x,v)` （其中 x 指这次操作的下标，v 指这次操作的权值），它执行以下代码：

```
void operate(int x,int v){
    for(;x<=n;x+=x & -x){
        machine[x]+=v;
    }
}
```

其中，我们把 `x+=x & -x` 叫做这个机器对于下标x的一次转递。

Sixy 固执地认为，这种操作特别像一种 OI 算法。

接着，Sixy 又发现了一串长为 m 的操作，这些操作的下标 x 是 add_i。幸运的是，这些操作的权值 v 都是 1。

Sixy 很高兴，他正好可以使用这个机器去执行这些操作。但突然，一束宇宙射线打中了这个机器，使这个机器产生一定程度的损坏。具体地，这个机器对于下标 x 的一次转递成功的概率变成了 p_x。当一次传递失败后，这个 `operate()` 操作将不再执行。

Sixy 很伤心，但他还是用这个机器执行了这些操作。现在他想知道，在这个机器中, $machine_x$ 的值恰好等于 k 的概率。

输入格式

第 1 行一个正整数 n，表示机器 $machine$ 的长度为 n。

第 2 到 $n+1$ 行每行两个整数 x, y，表示这个机器下标 i 一次传递成功的概率 $p_i = \dfrac{x}{y}$。

第 $n+2$ 行一个正整数 m, 表示这串操作的长为 m。

第 $n+3$ 行一行 m 个正整数 add_i，$1 \le i \le m$，表示每个操作的下标，**保证 add_i 各不相同。**

第 $n+4$ 行一行一个正整数 q,表示有 q 个询问。

第 $n+5$ 行到第 $n+q+4$ 行每行两个正整数 x, k，询问 $machine_x$ 的值恰好等于 k 的概率。

输出格式

对于每个询问，输出一行 $machine_x$ 的值恰好等于 k 的概率，对 998244353 取模。

所有运算都在模 998244353 的意义下。

样例

样例输入 #1

```
8
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
4
1 3 5 7
1
8 4
```

样例输出 #1

```
1
```

样例解释：每个点上传的概率都为 1,所以每个操作一定执行。

那对于第一个操作，$machine_1+=1$、$machine_2+=1$、$machine_4+=1$、$machine_8+=1$。

对于第二个操作，$machine_3+=1$、$machine_4+=1$、$machine_8+=1$。

对于第三个操作，$machine_5+=1$、$machine_6+=1$、$machine_8+=1$。

对于第四个操作， $machine_7+=1$、$machine_8+=1$。

所以，$machine_8=4$ 的概率为 1。

样例输入 #2

```
8
1 1
1 1
1 2
1 1
1 2
1 1
1 1
1 1
2
3 5
2
8 1
8 2
```

样例输出 #2

```
499122177
748683265
```

样例解释：$machine_3+=1$、$machine_5+=1$ 有 1 的概率执行，对于 3，执行一次传递的概率为 $\frac{1}{2}$。对于 5，执行一次传递的概率也为 $\frac{1}{2}$。

当 3 完成一次传递，会变成 4，当 5 完成一次传递，会变成 6，而当 $4,6$ 完成一次传递，都会变成 8，对于 $4,6$，有 1 的概率上传。

所以 $machine_4+=1$、$machine_6+=1$ 有 $\frac{1}{4}$ 的概率同时执行，也有 $\frac{1}{4}$ 的概率同时不执行，所以 $machine_8=2$ 的概率是 $\frac{1}{4}$，$machine_8=0$ 的概率也是 $\frac{1}{4}$。由于 $machine_8$ 一定小于 2，相减得 $machine_8=1$ 的概率是 $\frac{1}{2}$。

样例输入 #3

```
8
1 1
1 1
1 2
1 2
1 1
1 1
1 1
1 1
1
3
1
8 1
```

样例输出 #3

```
748683265
```

样例 #3 解释

对于 3，执行一次传递的概率为 $\frac{1}{2}$。假设上传成功，3 传递到 4，则 $machine_4 += 1$ 的概率为 $\frac{1}{2}$。对于 4，执行一次传递的概率也为 $\frac{1}{2}$，传递到 8，所以这个操作上传到 8 的概率，也就是 $machine_8 += 1$ 的概率是 $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$。

数据规模

测试点	n	m	q	特殊性质
1	$\le 2 \times 10^5$	$= 0$	$\le 5 \times 10^5$	无
2~4	$\le 2 \times 10^5$	≤ 300	$\le 5 \times 10^5$	所有上传的概率为1
5~7	≤ 10	≤ 10	≤ 2000	无
8	≤ 1000	$= 1$	$\le 5 \times 10^5$	询问中的v全是1
9~10	≤ 1000	$= 1$	$\le 5 \times 10^5$	无
11~25	$\le 2 \times 10^5$	≤ 300	$\le 5 \times 10^5$	无

对于所有测试点，$1 \le n \le 2 \times 10^5$、$1 \le m \le 300$、$1 \le q \le 5 \times 10^5$，所有数据在输入时会对 998244353 取模。