

题目名称	斐波那契	序列	栅栏
提交文件名	f.cpp	x.cpp	z.cpp
输入文件名	f.in	x.in	z.in
输出文件名	f.out	x.out	z.out
时间限制	2s	2s	2s
空间限制	512MB	512MB	512MB
题目类型	传统	传统	传统

注意事项：

1. 数据较大，注意使用读入优化。
2. 题目难度与题目顺序无关。
3. 编译不打开 O2 优化。

1 斐波那契

1.1 description

斐波那契数列又称兔子数列，可以通过以下方式产生：一开始只有一只兔子，一个月之后这只兔子每个月会繁殖出另一只兔子。之后每只兔子出生后都按照以上规则繁殖。我们把每个月的兔子的数量作为数列中的数就可以得到斐波那契数列。

现在草原上有 n 只兔子排成一排，每只兔子有一个繁殖能力 a_i 。小 F 想把这些兔子分组，要求一个组内的兔子必须是连续的一段。而且因为兔子们不太喜欢兔子数列，所以组内任意两只兔子的繁殖能力的和不能在兔子数列中出现。

现在小 F 想让这些兔子的分组数量尽可能少，但是兔子很多，他算不过来，于是他找到了你，想让你告诉他最少的分组数量是多少。

1.2 input

第一行一个整数 n ，表示兔子的个数。

接下来一行 n 个数，依次表示 a_i 。

1.3 output

输出一个数，表示最小的分组数量。

1.4 sample input

```
5
1 5 2 6 7
```

1.5 sample output

```
4
```

1.6 note

最优分组的一种为： $\{1\}, \{5, 2\}, \{6\}, \{7\}$ 。

1.7 data range

对于 10% 的数据， $n \leq 20$ 。

对于 30% 的数据， $n \leq 300$ 。

对于 60% 的数据， $n \leq 1000$ 。

对于 100 % 的数据， $n \leq 100000, a_i \leq 10^9$ 。

2 序列

2.1 description

最近小 X 对序列的研究产生了一定的兴趣。

小 X 把一个区间的中位数看成将这个区间内的数排序之后位于中间的数，如果区间的长度为偶数的话则是较小的数。

一开始，小 X 想研究一个序列的所有区间中，中位数最大的区间是哪一个。

后来他想了想觉得太简单了，于是决定把这个问题加大一点难度：小 X 现在想知道有多少区间的中位数位于他给定的 $[l_1, r_1]$ 之间。

后来他想了想便自己解决了这个问题，于是他再加入了一个限制：区间的长度必须位于 $[l_2, r_2]$ 之间。

小 X 发现加了这样一个限制之后，他自己也不会做了。于是他想让你帮他解决这个问题。

2.2 input

第一行有两个正整数 n, m ，表示序列的长度和询问的次数。

第二行有 n 个数，分别表示原来序列的数。

接下来有 m 行，每一行有四个正整数，分别表示每个询问的 l_1, r_1, l_2, r_2 。

2.3 output

输出 m 行，每一行表示每个询问对应的答案。

2.4 sample input

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

2.5 sample output

```
5
8
8
```

0

6

2.6 data range

对于 10% 的数据, $n \leq 100$ 。

对于 30% 的数据, $n \leq 1000$ 。

另有 40% 的数据, $l_2 = 1, r_2 = n$ 。

对于 100% 的数据, $n, a_i \leq 100000, m \leq 5$ 。

3 栅栏

3.1 description

小 Z 决定要去 AK PION 2018, 不过在这之前, 他要先把家里的牛给养大。

小 Z 家里的牛棚是一个 $n * m$ 的网格, 牛均匀地分布在这些网格中。

一开始小 Z 的牛都很小, 习性也都相近, 所以牛棚中没有栅栏。不过随着时间的推移, 不同品种的牛的不同习性逐渐暴露出来, 这时候小 Z 会建造栅栏来把不同习性的牛给分隔开。不过, 时间久了之后, 牛之间也需要交流感情, 所以小 Z 有时也可能拆除一些栅栏。

每次小 Z 建造栅栏时, 总会先圈出一个矩形内部的牛, 然后在这个矩形的边界围一圈栅栏, 也就是说一次建造的栅栏会将某个矩形的内部与外部给隔开。

每次拆除栅栏时, 小 Z 只会拆除之前某个时间建造的栅栏, 也就是说一次拆除只会拆除之前某个时间建造的一个矩阵边界上的所有栅栏。

不过问题也来了: 家里的牛棚太大了, 栅栏多了之后, 小 Z 根本不知道两个位置的牛是否被栅栏隔开, 所以小 Z 想让你帮他解决这个问题。

当然, 为了方便拆除和建造, 小 Z 会保证任意时刻栅栏与栅栏之间互不相交。

3.2 input

第一行两个整数 n, m, q 。 n, m 表示牛棚的长宽, q 表示接下来发生的事件数。

接下来 q 行, 每行有五个整数 p, x_1, y_1, x_2, y_2 。

如果 $p = 1$, 表示小 Z 在以 (x_1, y_1) 为左上角, (x_2, y_2) 为右下角的矩形边界处建造了栅栏。

如果 $p = 2$, 表示小 Z 要拆除以 (x_1, y_1) 为左上角, (x_2, y_2) 为右下角的矩形外的栅栏 (保证存在)。

如果 $p = 3$, 表示小 Z 想询问你位于 $(x_1, y_1), (x_2, y_2)$ 的牛是否被栅栏隔开。

3.3 output

对于每个询问, 如果两个位置没有被隔开, 则输出 "Yes", 否则输出 "No"。

3.4 sample input

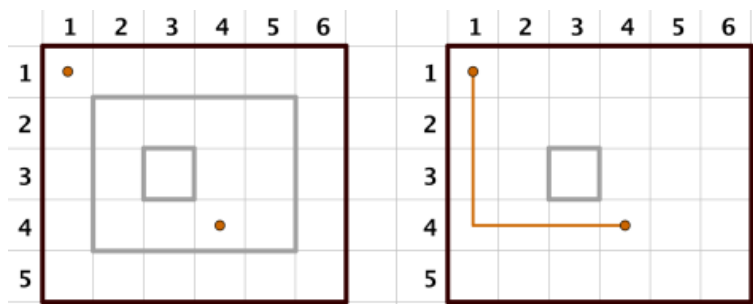
```
5 6 5
1 2 2 4 5
1 3 3 3 3
3 4 4 1 1
2 2 2 4 5
3 1 1 4 4
```

3.5 sample output

No

Yes

3.6 note



3.7 data range

对于 20% 的数据, $n, m, q \leq 300$ 。

对于 40% 的数据, $n, m, q \leq 2000$ 。

另有 20% 的数据, $n = 1$ 。

另有 20% 的数据, $q \in \{1, 3\}$, 且所有 3 操作在 1 操作之后。

对于 100% 的数据, $n, m \leq 2000, q \leq 100000$ 。