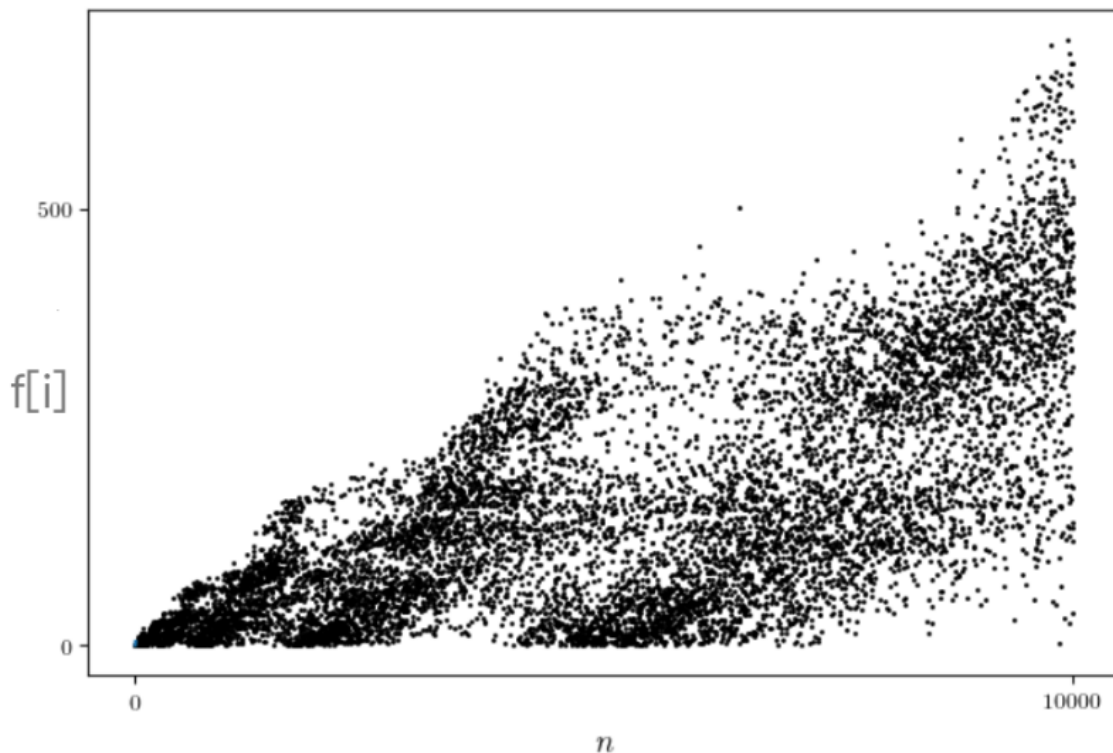


数列(seq.cpp/in/out)

Cuber QQ 痴迷于对数列的研究。最近他发现了一个映射酷似火焰的数列。



这个数列 $f_0, f_1, f_2, \dots, f_n$, 其中 $f_0 = 1, f_1 = 1$ 。

而且 f_i 是满足对于任意的正整数 $k(k > 0, i - 2k \geq 0)$ 一定有 $f_i - f_{i-k} \neq f_{i-k} - f_{i-2k}$ 的最小的正整数。

现在 Cuber QQ 会给你一个整数 n , 希望你求出 f_n 是多少。

输入

输入包含一行一个整数 $n(1 \leq n \leq 1000)$ 。

输出

输出包含一行一个整数, 表示 f_n 。

样例 1

输入

5

输出

2

样例 2

输入

8

输出

4

数据范围约定

对于其中 9% 的数据保证 $n \leq 100$

对于其中 100% 的数据保证 $n \leq 1000$

时间限制：2 s

空间限制：1024 MB

最小公倍数(lcm.cpp/in/out)

Cuber QQ 喜欢最小公倍数。

两个或多个整数公有的倍数叫做它们的公倍数，其中除 0 以外最小的一个公倍数就叫做这几个整数的最小公倍数。

我们经常用质因数分解法来求最小公倍数：把每个数分别分解质因数，再把各数中的全部公有质因数提取出来连乘，所得的积就是这几个数的最大公约数。

Cuber QQ 自己刚刚做了一道题目：三个小朋友经常去图书馆，分别每隔 3 天、2 天、5 天去图书馆借书，他们恰好在第一天的时候都去了图书馆，要让 Cuber QQ 求出他们最少过几天又会在同一天去图书馆。这个问题对已经会了最小公倍数的 Cuber QQ 来说很简单。

Cuber QQ 决定将问题升级，Cuber QQ 现在会召集一些小朋友，他们会分别间隔一定的天数规律地来到图书馆，而一旦某一天图书馆来了至少一个小朋友，Cuber QQ 都会告诉你这天的日期。Cuber QQ 会在第一天的时候召集所有人来到图书馆。他现在想知道，对于他给定的日期，他最少需要召唤多少小朋友。

Cuber QQ 只会记录在 $2 \cdot 10^5$ 以内出现小朋友的日期，也就是说，后续的日期你可以忽略不计。**当然 Cuber QQ 也可以在任意时刻结束记录。**

Cuber QQ 的记录当然是不会出错的，也就是输入的数据保证存在合法的解。

输入

输入数据第一行包含一个整数 $n(1 \leq n \leq 2 \cdot 10^5)$ ，表示有其中 n 天是有人来的。

第二行包含 n 个用空格分开的整数 $a_1, a_2, \dots, a_n(1 = a_1 < a_2 < \dots < a_n \leq 2 \cdot 10^5)$ ，表示 n 个人来的日期。

输出

输出一行包含一个整数，表示需要的最少小朋友数量。

样例 1

输入

```
3
1 2 3
```

输出

```
1
```

解释

我们只需要一个小朋友，每天都来。

样例 2

输入

```
4
1 4 5 7
```

输出

```
2
```

解释

我们至少需要两个小朋友，分别间隔 3、4 天来。

数据范围约定

对于 7% 的数据，满足 $n \leq 20$

对于 18% 的数据，满足 $n \leq 300$

对于 40% 的数据，满足 $m \leq 5000$

对于 100% 的数据，满足 $n \leq 2 \cdot 10^5$

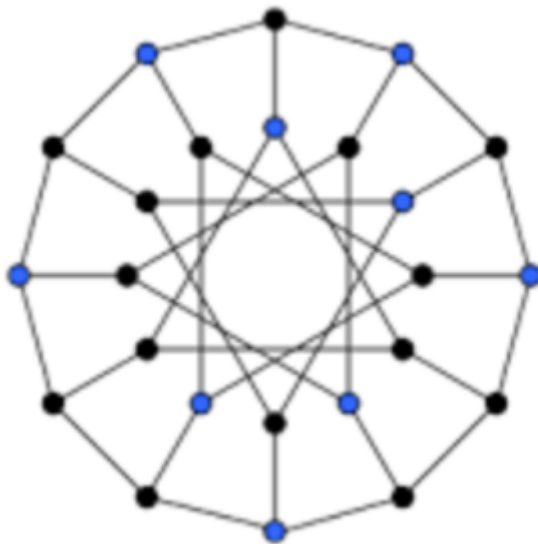
时间限制：2 s

空间限制：1024 MB

独立集(set.cpp/in/out)

Cuber QQ 对图论造诣颇深。

最近 Cuber QQ 在研读独立集相关的论文。



独立集是一个图论中的概念，是指图 G 中两两互不相邻的顶点构成的集合，例如上图蓝点就是该图的一个独立集。

而大家都知道，任意有关图中团的性质都能很自然的转述成独立集的性质。一般而言，寻找图的最大团是 NP 困难的，从而寻找图的最大独立集也是 NP 困难的。

于是 Cuber QQ 觉得最大独立集一个值得思考的问题。最大独立集的概念很容易从独立集得到，最大独立集就是图 G 中两两顶点之间都不相连的 **最多** 个数的集合。

不过，现在 Cuber QQ 想求的是，图 G 中所有子图的最大独立集之和。

如果图 $G'(V', E')$ 是图 $G(V, E)$ 的子集（ V 和 E 分别表示点的集合和边的集合），满足：

- $V' \subseteq V$
- 边 $(a, b) \in E'$ 当且仅当 $a \in V', b \in V'$ 而且边 $(a, b) \in E$ 。

输入

输入第一行包含两个整数 $n, m (2 \leq n \leq 26, 0 \leq m \leq \frac{n \times (n-1)}{2})$ ，分别表示图 G 的点数和边数。

接下来 m 行，每行两个整数 $x_i, y_i (0 \leq x_i < y_i < n)$ 表示图 G 中的边。

图 G 中的点的编号为 $0, 1, 2, \dots, n-1$ 。

保证图中不存在自环和重边。

输出

一行一个整数，表示图 G 中所有子图的最大独立集之和。

样例 1

输入

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

输出

```
9
```

解释

子图的最大独立集大小分别为：

- $\{\}$: 0
- $\{0\}$: 1
- $\{1\}$: 1
- $\{2\}$: 1
- $\{0, 1\}$: 1
- $\{0, 2\}$: 1
- $\{1, 2\}$: 2
- $\{0, 1, 2\}$: 2

$0 + 1 + 1 + 1 + 1 + 1 + 2 + 2 = 9$ 。

样例 2

输入

```
7 5
0 5
3 4
1 2
2 5
0 2
```

输出

```
328
```

样例 3

输入

```
18 12
6 13
7 5
3 16
16 14
0 3
3 11
13 7
13 15
9 17
2 5
10 13
14 10
```

输出

```
1868608
```

数据范围约定

对于 12% 的数据, 满足 $n \leq 5$

对于 60% 的数据, 满足 $n \leq 18$

对于 100% 的数据, 满足 $n \leq 26$

时间限制: 2 s

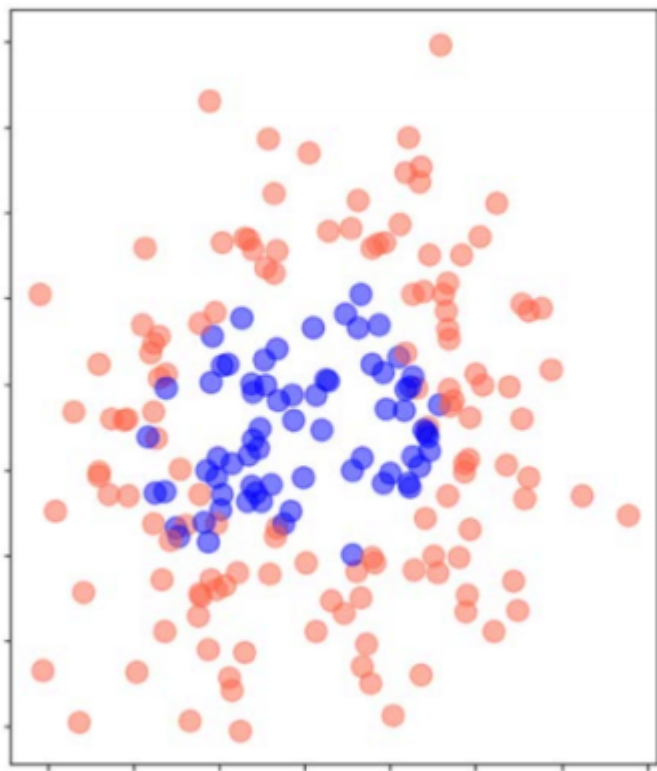
空间限制: 1024 MB

聚类(class.cpp/in/out)

Cuber QQ 最近在研究聚类。

聚类是一种包括数据点分组的机器学习技术。给定一组数据点, 我们可以用聚类算法将每个数据点分到特定的组中。理论上, 属于同一组的数据点应该有相似的属性和/或特征, 而属于不同组的数据点应该有非常不同的属性和/或特征。聚类是一种无监督学习的方法, 是一种在许多领域常用的统计数据分析技术。

Cuber QQ 当然不会拿非常专业的学术问题来考你, 现在 Cuber QQ 已经将平面上的点分成了两类, 分别标记为蓝色和红色。



Cuber QQ 认为蓝色的点是十分重要的, 所以每一个蓝色点都拥有价值 A ; 而红色的点是十分碍眼的, 所以每一个红色点都拥有价值 $-B$ 。

现在 Cuber QQ 想让你解决的问题是, 在这个平面上选择一个矩形区域 (矩形必须与坐标轴平行), 使得这个矩形区域内的价值和最大, 即选定的矩形内 (包括边界上) 点的价值总和最大。

输入

输入数据第一行包含一个整数 n ($1 \leq n \leq 1000$), 表示蓝点的数量。

接下来的 n 行, 每行两个整数 x_i, y_i ($0 \leq |x_i|, |y_i| \leq 10^9$), 表示蓝点的坐标。

接下来一行一个整数 m ($1 \leq m \leq 1000$), 表示红点的数量。

接下来的 m 行, 每行两个整数 $x_i, y_i (0 \leq |x_i|, |y_i| \leq 10^9)$, 表示红点的坐标。

最后一行两个整数 $A, B (1 \leq A, B \leq 10000)$ 表示点的价值, 如题面所述。

输出

输出一行包含一个整数, 表示答案。

样例1

输入

```
2
-1 -1
4 4
2
0 0
2 2
5 2
```

输出

```
6
```

样例2

输入

```
3
0 5
3 3
8 -1
3
1 4
6 0
7 1
3 2
```

输出

```
4
```

样例3

下发样例3

数据范围约定

对于其中 42% 的数据, 保证 $1 \leq n, m \leq 20$ 。

对于其中 66% 的数据, 保证 $1 \leq n, m \leq 500$ 。

对于其中 100% 的数据, 保证 $1 \leq n, m \leq 1000$ 。

时间限制: 2 s

空间限制: 1024 MB