

# 冒泡排序

*qiu.cpp/qiu.in/qiu.out*

*TimeLimit : 1000ms    MemoryLimit : 256MB*

## 题目背景

冒泡排序 (**Bubble Sort**)，是一种计算机科学领域的较简单的排序算法。

冒泡排序算法的原理如下：

1. 比较相邻的元素。如果第一个比第二个大，就交换他们两个。
2. 对每一对相邻元素做同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。
3. 针对所有的元素重复以上的步骤，除了最后一个。
4. 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

## 题目描述

给出  $n$  个数  $a_i$ ，请你求出将其按照冒泡排序的规则从小到大排序需要的交换次数。

## 输入格式

第一行一个正整数  $n$ 。

第二行  $n$  个正整数  $a_{1..n}$

## 输出格式

一行，输出  $n$  个数，每个数之间用一个空格分隔。

## 样例输入1：

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

## 样例输出1：

```
1 4
```

样例解释：

第一轮：

第一次交换：3 2 1 2 4 → 2 3 1 2 4

第二次交换：2 3 1 2 4 → 2 1 3 2 4

第三次交换：2 1 3 2 4 → 2 1 2 3 4

第二轮：

第四次交换：2 1 2 3 4 → 1 2 2 3 4

此时完成了对序列的从小到大的排序。

样例输入2:

```
1 6
2 9 17 5 3 1 20
```

样例输出2:

```
1 9
```

数据范围

对于前 20% 的数据,  $1 \leq n \leq 5000$ 。

对于前 60% 的数据,  $\forall 1 \leq i \leq n, 1 \leq a_i \leq 100$ 。

对于前 100% 的数据,  $1 \leq n \leq 10^5, \forall 1 \leq i \leq n, 1 \leq a_i \leq 10^6$ 。

# 航线规划

*route.cpp/route.in/route.out*

*TimeLimit : 2000ms    MemoryLimit : 512MB*

## 题目描述

没有机场的 A 市经济逐渐发达，市长为了促进发展，拟修建 $n$ 个机场。为了避免飞机在空中相遇导致灾难，市长在这 $n$ 个机场之间，规划了 $m$ 条航线。一条航线可以由一对机场描述，意味着飞机可以由其中任意一个机场飞往另一个机场。但是，飞机飞行也需要时间，所以每条航线有一个权值，意味着经过这条航线需要多少时间。

一条航线是可以被通行的，当且仅当，它的两个机场都已经被修建好了。

当然，机场的修建是需要时间的。偶尔，施工队会告诉市长，有某一个机场被修好了。而在机场修建的同时，市长为了监督进度，会想要知道从某一个已修建的机场到另一个已修建的机场，仅通过飞机，最快需要多久。

而你，作为市长秘书，当然要负责回答，在已有的机场条件下，市长的疑问。

特殊的，如果两个机场之间，没有任何一条路径能互相到达，你会回答市长“-1”。

## 输入格式

输入第一行为两个非负整数， $n$ 、 $m$ 、 $q$ ，分别表示机场的数量和航线的数量和你所需要处理的操作数。

接下来共有 $m$ 行来描述所有的航线。其中每行都有三个正整数， $u$ 、 $v$ 、 $w$ ，分别表示一条航线的两个机场和经过它所需的时间。

最后有 $q$ 行，每行第一个数 $opt$ ，为操作类型：如果 $opt = 1$ ，意味着这是施工队报告一个机场被修好了，那么接下来会有一个正整数 $x$ ，其中 $x \leq n$ ，表示被修好的机场；如果 $opt = 2$ ，意味着市场询问了你，那么接下来会有两个正整数 $x$ 、 $y$ ，其中 $x$ 、 $y \leq n$ ，表示你要回答从 $x$ 到 $y$ 在已有机场下，最快需要多久。

**数据保证每个机场至多被修好一次；市长询问的城市总是已经修好的城市，但不一定是两个不同的城市。**

## 输出格式

对于 $q$ 个操作中每一个 $opt = 2$ 类型输出一个值，表示最快时间。当然，如果到不了的话，你要输出-1。

## 样例输入:

```
1 4 4 6
2 1 2 1
3 2 3 2
4 1 3 4
5 1 4 5
6 1 1
7 1 4
8 2 1 4
9 1 3
10 1 2
11 2 1 3
```

样例输出:

```
1 5
2 3
```

### 数据范围

对于前20%的数据, 保证 $q \leq 3$ ;

对于前50%的数据, 保证 $n \leq 100$ ;

对于全部的数据, 保证 $n \leq 300$ ,  $m \leq \frac{n(n-1)}{2}$ ,  $q \leq 10^6$ ,  $w_i \leq 10^6$ 。

# 数字谜题

`number.cpp/number.in/number.out`

`TimeLimit : 1000ms`    `MemoryLimit : 512MB`

## 题目描述

你获得了两个长度均为  $n$  的数字序列  $a, b$  , 满足  $\forall i \in [1, n], a_i, b_i \in [0, n]$ 。现在你需要解决一个数字谜题, 你需要对于一个给定的  $m \in [0, n)$  , 求出有多少个有序二元组  $(i, j), i, j \in [1, n]$  满足  $a_i \times b_j \bmod n = m$

## 输入格式

第一行两个整数  $n, m$

第二行一行  $n$  个整数, 第  $i$  个整数为  $a_i$

第三行一行  $n$  个整数, 第  $i$  个整数为  $b_i$

## 输出格式

一行一个整数, 表示答案

## 样例输入

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

## 样例输出

```
1 4
```

## 数据范围

对于前 30% 的数据, 满足  $n \leq 1000$

对于另外 30% 的数据, 满足  $n = 99991$  (99991是一个质数)

对于 100% 的数据, 满足  $n \leq 2 \times 10^5$

# 游戏

*game.cpp/game.in/game.out*

*TimeLimit : 2000ms    MemoryLimit : 512MB*

## 题目描述

小P是个喜欢矩阵的女孩子。

她现在有一个 $n \times n$ 的，元素均为0或1的矩阵，并打算对它做些操作。

但是如果操作很普通，矩阵就会是静态的，很不美观。

小P还喜欢康威生命游戏。因此她规定：

- 一、如果该矩阵一个位置的元素为1，这个位置就有一个活细胞。否则，这个位置有一个死细胞。
- 二、令初始时刻为0，每过一秒，这个矩阵中的细胞的生死情况都会改变。对于一个细胞，在以它为中心的八联通块的四个角上的细胞中，有奇数个是存活的，则该细胞下一秒的生死情况为生，否则为死。
- 三、矩阵范围外没有细胞，也可以认为矩阵外的细胞始终是死的。

注意下一秒细胞的生死关系与当前时刻他自己的生死情况无关，上述规则与康威生命游戏也并不相同。

小P现在把这个矩阵给了你，她想知道，经过 $t$ 秒后，这个矩阵每个位置的细胞的生死情况是怎样的。

## 输入格式

第一行两个正整数 $n, t$ ，表示矩阵长宽大小与经过的秒数 $t$ 。

接下来 $n$ 行，每行 $n$ 为0或1的数，表示每个元素的取值，即每个位置细胞生死情况。

## 输出格式

一个 $n \times n$ 的矩阵，表示 $t$ 秒后细胞的生死情况。1表示生，0表示死。不需要用空格分隔。

## 样例输入1

```
1 5 1
2 11001
3 10110
4 11101
5 10111
6 10010
```

## 样例输出1

```
1 00111
2 01010
3 00010
4 11001
5 00101
```

## 样例解释1

这个游戏共进行了一轮。

对于位置(1,2)，对他有影响的位置为(2,1)与(2,3)，这两个位置细胞均为活，存活个数为偶数，因此1秒后(1,1)细胞为死。

对于位置(2,4)，对他有影响的位置为(1,5)，(1,3)，(3,5)与(3,3)，有三个位置细胞为活，存活个

数为奇数，因此1秒后(2,4)细胞为活。  
其余位置类似。

## 样例输入2

```
1 6 1000000
2 101000
3 001001
4 111100
5 010110
6 101001
7 010100
```

## 样例输出2

```
1 011000
2 101000
3 000001
4 100011
5 010001
6 011010
```

## 数据范围

对于 10% 的数据， $n \leq 50, T \leq 10$ 。  
对于 30% 的数据， $n \leq 100, T \leq 100$ 。  
对于另外 10% 的数据， $n \leq 2, T \leq 10^9$ 。  
对于 90% 的数据， $n \leq 200, T \leq 10^9$ 。  
对于 100% 的数据， $n \leq 400, T \leq 10^9$ 。