

# 8102 PIgēON 模拟赛

2018 年 8 月

By applese

题目名称	冒泡排序	情报中心	百鸽笼
题目类型	传统型	传统型	传统型
目录	inverse	center	pigeon
可执行文件名	inverse	center	pigeon
输入文件名	inverse.in	center.in	pigeon.in
输出文件名	inverse.out	center.out	pigeon.out
每个测试点时限	1.0 秒	1.5 秒	2.0 秒
测试点数目	20	20	20
每个测试点分值	5	5	5
内存限制	256MB	256MB	256MB

提交源程序文件名

对于 C++ 语言	inverse.cpp	center.cpp	pigeon.cpp
对于 C 语言	inverse.c	center.c	pigeon.c
对于 Pascal 语言	inverse.pas	center.pas	pigeon.pas

编译选项

对于 C++ 语言	-lm -O2	-lm -O2	-lm -O2
对于 C 语言	-lm -O2	-lm -O2	-lm -O2
对于 Pascal 语言	-O2	-O2	-O2

注意事项

- 1、发现原题请不要声张，提前 AK 也请不要声张。
- 2、请不要使用 `#pragma` 等不能在 PIgēON 中使用的技巧。
- 3、请不要使用超出 PIgēON 可使用范围的函数、库等。
- 4、请对 XJ 评测机的速度有足够的信仰。
- 5、在 XJOI 上提交时，请使用标准输入输出流。
- 6、这是一套送温暖题。



## 1 冒泡排序

### 1.1 题目背景

最近，小 S 对冒泡排序产生了浓厚的兴趣。为了问题简单，小 S 只研究对 1 到  $n$  的排列的冒泡排序。

小 S 发现小 S 在 ION 赛场上居然写错了冒泡排序，很是生气。他教给了小 S 另一种冒泡排序。下面是对冒泡排序的算法描述。

---

#### 算法 1 冒泡排序

---

输入：一个长度为  $n$  的排列  $P$

输出： $P$  排序后的结果。

```
1: function SORT( $n, P$ )
2:   while  $P$  无序 do
3:      $x \leftarrow \text{RAND}(1, n)$ 
4:      $y \leftarrow \text{RAND}(1, n)$ 
5:     SWAP( $P[x], P[y]$ )
6:   end while
7:   return  $P$ 
8: end function
```

---

冒泡排序的交换次数被定义为交换过程的执行次数。

### 1.2 题面描述

小 S 开始专注于研究长度为  $n$  的排列，他想知道，在你运气足够好的情况下（即每次冒泡排序的交换次数都是可能的最少交换次数，仿佛有上帝之手在操控），对于一个等概率随机的长度为  $n$  的排列，进行这样的冒泡排序的期望交换次数是多少？

### 1.3 输入格式

从文件 `inverse.in` 中读入数据。

输入第一行包含一个正整数  $T$ ，表示数据组数。

对于每组数据，第一行有一个正整数，保证  $n \leq 10^7$ 。

### 1.4 输出格式

输出到文件 `inverse.out` 中。

输出共  $T$  行，每行一个整数。

对于每组数据，输出一个整数，表示答案对 998244353 取模的结果。



### 1.5 样例 1

#### 【样例 1 输入】

2  
2  
4

#### 【样例 1 输出】

499122177  
415935149

#### 【样例 1 解释】

当  $n = 2$  时，有两种可能的排列。

当排列为  $\boxed{1\ 2}$  时，交换次数为 0。

当排列为  $\boxed{2\ 1}$  时，交换次数为 1。

所以期望交换次数为  $\frac{1}{2}$ 。

### 1.6 样例 2

见选手目录下的 *inverse/inverse2.in* 与 *inverse/inverse2.ans*。

### 1.7 子任务

测试点编号	$T$	$n$
1 - 2	$\leq 10$	$\leq 10$
3 - 6	$= 1$	$\leq 10^7$
7 - 10	$\leq 10^5$	$\leq 10^3$
11 - 20	$\leq 10^5$	$\leq 10^7$

对于所有数据，满足  $1 \leq T \leq 10^5, 1 \leq n \leq 10^7$ 。

KEEP CALM AND THINK



## 2 情报中心

### 2.1 题目背景

- 。飞纷火战来年近国 D 和国 C
- 。飞乱子鸽来年近国 D 和国 C

### 2.2 题面描述

最近, C 国成功地渗透进入了 D 国的一个城市。这个城市可以抽象成一张有  $n$  个节点, 节点之间有  $m$  条双向道路连接的无向图, 每条道路的长度都为 1。

经过侦查, C 国情报部部长 GGB 惊讶地发现, 这座看起来不起眼的城市竟然是 D 国的军事中心。因此 GGB 决定在这个城市内设立情报机构。情报专家 TAC 在侦查后, 安排了  $q$  种设立情报机构的方案。这些方案中, 第  $i$  种方案将计划建立  $k_i$  个情报机构, 第  $j$  个情报机构可以安排人员到距离其不超过  $d_{i,j}$  的节点上收集情报。

但是, 由于人手不足, GGB 只能安排上述  $q$  种方案中的一种进行实施。为了评估一种方案的性能, 我们把能够收集到情报的节点数量视为这种情报的价值。现在, 小  $\infty$  被 GGB 和 TAC 派来侦查, 请你帮他统计每一种方案的价值。

### 2.3 输入格式

从文件 `center.in` 中读入数据。

输入第一行包含三个正整数  $n, m, q$ , 分别表示城市的节点个数、道路条数和方案个数。

接下去  $m$  行每行两个正整数  $u, v$ , 表示存在一条连接城市  $u$  和城市  $v$  的双向道路。

接下去  $q$  行, 每行表示一个方案。第一个正整数  $k$  表示该种方案将计划建立  $k$  个情报机构, 之后是  $2k$  个正整数, 其中第  $2i - 1$  个数表示方案中第  $i$  个情报机构所在的节点编号, 第  $2i$  个数表示第  $i$  个情报点所能派出情报人员的最远距离。

### 2.4 输出格式

输出到文件 `center.out` 中。

输出包含  $q$  行, 每行包含一个整数, 表示相应询问的答案。

### 2.5 样例 1

【样例 1 输入】

```
5 8 3
1 2
1 3
1 4
```



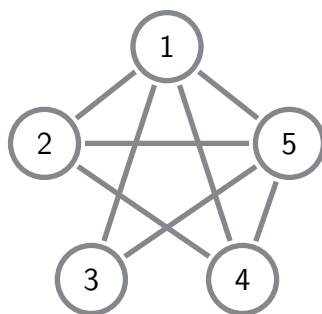
```
1 5
2 4
2 5
3 5
4 5
1 2 1
1 1 1
2 2 2 3 1
```

**【样例 1 输出】**

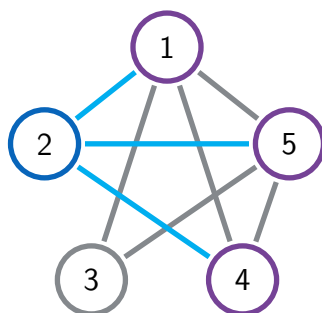
```
4
5
5
```

**【样例 1 解释】**

样例中图如下所示：



第一种方案如下所示：



其中，**蓝色节点**表示情报机构的位置，**紫色节点**表示可以安排情报人员的节点。

## 2.6 样例 2

见选手目录下的 `center/center2.in` 与 `center/center2.ans`。



## 2.7 子任务

测试点编号	$n$	$q$	$\sum k$	特殊性质
1 - 2	10	$\leq 10$	$\leq 30$	无
3	200	$\leq 200$	$\leq 2000$	图构成一棵树
4 - 6	200			无
7	598	$\leq 600$	$\leq 10000$	图构成一棵树
8	599			图构成一条链
9 - 12	600			无
13	998	$\leq 1000$	$\leq 2 \times 10^6$	图构成一棵树
14	999			图构成一条链
15 - 20	1000			无

对于所有数据, 保证  $0 < n, q \leq 1000$ ,  $0 < m \leq \min(10^5, \frac{n(n-1)}{2})$ ,  $0 < \sum k \leq 2 \times 10^6$ ,  $1 \leq u, v \leq n$ 。

KEEP CALM AND CODE



## 3 百鸽笼

### 3.1 题目背景



### 3.2 题面描述

在 JOU 管理员群里一共有  $n$  个管理员，为了容纳这些管理员， $vfk$  准备了  $n$  个鸽笼。每个鸽笼中都装有一个咕咕能力值为  $v_i$  的鸽子，每只鸽子能力值不一定相同。每当 RU 开始或结束或咕咕咕时，管理员们就会对这些鸽笼进行操作。操作包括三种：

- 1  $vfk$  和管理员们取出最左端的鸽笼。
- 2  $v$   $vfk$  和管理员们在最左侧新放入一个装有咕咕能力值为  $v$  的鸽子的鸽笼。
- 3  $l\ r\ k$   $vfk$  想要知道从左到右第  $l$  个到第  $r$  个鸽笼中鸽子的第  $k$  小的咕咕能力值。

小  $\infty$  想要知道每个 3 操作（即询问）的答案，请你来帮帮他。

### 3.3 输入格式

从文件 *pigeon.in* 中读入数据。

输入第一行包含两个正整数  $n, m$ ，分别表示初始鸽笼数与操作个数。

第二行包含  $n$  个正整数，第  $i$  个数表示从左往右第  $i$  个初始鸽笼中鸽子的咕咕能力值  $v_i$ 。

接下去  $m$  行每行表示一个操作。操作输入格式见题面描述。

### 3.4 输出格式

输出到文件 *pigeon.out* 中。

输出包含若干行，每行表示一个相应的 3 操作的答案。

3.5 样例 1

【样例 1 输入】

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

【样例 1 输出】

5  
4  
6  
9

【样例 1 解释】

初始序列如下：



第一次操作，查询区间 [2, 5]：



其中橙色鸽子表示在询问区间内的鸽子。其中第 3 小的咕咕能力值为 5。

第二次操作后：



第三次操作后：



其中蓝色鸽子表示新加入的鸽子。

第四次操作，查询区间 [1, 4]：





第 2 小的咕咕能力值为 4。

3.6 样例 2

见选手目录下的 `pigeon/pigeon2.in` 与 `pigeon/pigeon2.ans`。

3.7 子任务

测试点编号	$n$	$m$	特殊性质
1 - 4	$\leq 2000$	$\leq 2000$	无
5	$\leq 3 \times 10^4$	$\leq 3 \times 10^4$	$k$ 均为 1 或询问区间长度
6 - 8			无
9	$\leq 8 \times 10^4$	$\leq 8 \times 10^4$	$k$ 均为 1 或询问区间长度
10			没有 1 操作
11			没有 2 操作
12 - 13			无
14	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$k$ 均为 1 或询问区间长度
15			没有 1 操作
16			没有 2 操作
17 - 20			无

对于所有数据，保证  $1 \leq n \leq 2 \times 10^5$ ， $1 \leq m \leq 2 \times 10^5$ 。保证数据合法，即  $1 \leq k \leq r - l + 1$ 。

KEEP CALM AND DEBUG

KEEP CALM AND GET ACCEPTED