

# NOIP 2019 模拟赛 Contest 5

diamond\_duke

题目名称	染色	乘方	位运算
可执行文件名	color	power	bitop
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	3s	1s
内存限制	512MB	512MB	512MB
子任务个数	4	5	6
题目类型	传统型	传统型	传统型

**请注意：** 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

# 1 染色

## 1.1 Problem Statement

小 D 正在研究染色。

小 D 得到了一张  $n$  个点  $m$  条边的无向图,它的所有节点被依次编号为  $1, 2, \dots, n$ 。

初始时,所有边都是白色的,而小 D 可以选择一些边将它们染成黑色。

在**任意时刻**,如果存在一个节点  $u$ ,满足所有与  $u$  相连的边中,有且仅有一条边是白色的,那么这条边也会变为黑色。值得一提的是,这里我们认为一条边同时给两个端点各贡献了一条边的计数,因此**自环会被认为是两条边**。

小 D 希望能够在上述过程无法再次发生时,所有边都是黑色的。

因为染色很累,所以小 D 想要最小化初始时染成黑色的边的个数。

但是小 D 并不会,请你帮帮他。

## 1.2 Input Format

从标准输入读入数据。

第一行两个正整数  $n, m$ , 表示无向图的点数以及边数。

接下来  $m$  行,每行两个空格隔开的整数  $u, v$ , 表示图中存在一条连接节点  $u, v$  的边。

## 1.3 Output Format

向标准输出输出答案。

输出一行一个整数,表示最小的染色边数。

## 1.4 Sample 1

### 1.4.1 Input

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

### 1.4.2 Output

```
1
```

### 1.4.3 Explanation

任意染其中的一条边都是一组可行的最优解。

### 1.5 Sample 2

见下发文件 `color/color2.in` 与 `color/color2.ans`。

### 1.6 Sample 3

见下发文件 `color/color3.in` 与 `color/color3.ans`。

### 1.7 Constraints

对于所有测试数据,  $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 2 \times 10^5$ ,  $1 \leq u, v \leq n$ , 图中可能有自环或重边。

- 子任务 1 (15 分):  $n, m \leq 20$ ;
- 子任务 2 (20 分):  $n, m \leq 5000$ ;
- 子任务 3 (25 分):  $n \leq 5000$ ;
- 子任务 4 (40 分): 无特殊限制。

## 2 乘方

### 2.1 Problem Statement

小 D 正在研究乘方。

小 D 定义集合  $S(t)$  表示所有正整数的  $t$  次幂组成的集合。形式化地,  $S(t) = \{n^t \mid n \in \mathbb{Z}^+\} = \{1^t, 2^t, 3^t, \dots\}$ 。

小 D 想要研究一些幂次的性质。因此, 小 D 会提出  $q$  个问题, 每个问题会给出  $k$  个数  $n_1, n_2, \dots, n_k$ , 而小 D 想要知道不可重集  $\bigcup_{i=1}^k S(n_i)$  中第  $m$  小的数字, 其中  $\bigcup$  指集合并运算。

但是小 D 并不会, 请你帮帮他。

### 2.2 Input Format

从标准输入读入数据。

第一行一个正整数  $q$ , 表示问题的个数。

接下来  $2q$  行, 每相邻两行表示一个问题:

- 第一行中有两个空格隔开的整数  $m, k$ 。
- 第二行中有  $k$  个空格隔开的整数  $n_1, n_2, \dots, n_k$ 。

### 2.3 Output Format

向标准输出输出答案。

输出  $q$  行, 每行一个整数, 表示  $\bigcup_{i=1}^k S(n_i)$  中第  $m$  小的数字。

### 2.4 Sample 1

#### 2.4.1 Input

```
1
12 3
2 3 5
```

#### 2.4.2 Output

```
81
```

### 2.5 Sample 2

见下发文件 `power/power2.in` 与 `power/power2.ans`。

## 2.6 Sample 3

见下发文件 `power/power3.in` 与 `power/power3.ans`。

## 2.7 Constraints

对于所有测试数据,  $1 \leq q \leq 1000$ ,  $1 \leq m \leq 10^9$ ,  $1 \leq k \leq 50$ ,  $1 \leq n_i \leq 50$ , 所有  $n_i$  互不相同, 保证答案不超过  $10^{17}$ 。

- 子任务 1 (15 分):  $m \leq 200$ ;
- 子任务 2 (20 分):  $m \leq 5000$ ;
- 子任务 3 (30 分):  $k \leq 10$ ;
- 子任务 4 (20 分):  $q \leq 10$ ;
- 子任务 5 (15 分): 无特殊限制。

## 3 位运算

### 3.1 Problem Statement

小 D 正在研究位运算。

众所周知，常见的二元位运算有如下三种：

1. **按位与**（即 C++ 中的 `&`），记作  $x \text{ and } y$ ；
2. **按位异或**（即 C++ 中的 `^`），记作  $x \text{ xor } y$ ；
3. **按位或**（即 C++ 中的 `|`），记作  $x \text{ or } y$ ；

小 D 想了一个长度为  $n$  的序列  $a_1, a_2, \dots, a_n$  和一个位运算符  $\text{op}$ 。

小 D 想要知道，如果他在序列中任取两个元素  $a_i, a_j$ ，并对他们使用运算符  $\text{op}$ ，可能得到的最大结果。形式化地，小 D 想要计算下式的值：

$$\max_{1 \leq i < j \leq n} \{a_i \text{ op } a_j\} \quad (1)$$

此外，小 D 还想要知道，有多少对  $(i, j)$  可以使得上式取到最大值。

但是小 D 并不会，请你帮帮他。

### 3.2 Input Format

从标准输入读入数据。

第一行两个正整数  $n, q$ ，其中  $n$  表示序列长度， $q$  表示操作符类型：

- 若  $q = 1$ ，则操作符  $\text{op}$  为**按位与**运算（and）；
- 若  $q = 2$ ，则操作符  $\text{op}$  为**按位异或**运算（xor）；
- 若  $q = 3$ ，则操作符  $\text{op}$  为**按位或**运算（or）；

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示序列中的元素。

### 3.3 Output Format

向标准输出输出答案。

输出一行两个整数，表示小 D 要求出的最大值以及取到最大值的  $(i, j)$  个数。

### 3.4 Sample 1

#### 3.4.1 Input

```
5 1
1 4 5 7 9
```

### 3.4.2 Output

5 1

## 3.5 Sample 2

### 3.5.1 Input

5 2  
2 3 4 5 7

### 3.5.2 Output

7 2

## 3.6 Sample 3

### 3.6.1 Input

5 3  
1 2 4 5 9

### 3.6.2 Output

13 2

## 3.7 Sample Explanations

- 对于第一组样例，唯一的最大值 5 在  $(i, j) = (3, 4)$  时取到。
- 对于第二组样例，最大值 7 在  $(i, j) = (1, 4)$  或  $(2, 3)$  时取到。
- 对于第三组样例，最大值 13 在  $(i, j) = (3, 5)$  或  $(4, 5)$  时取到。

## 3.8 Constraints

对于所有测试数据， $2 \leq n \leq 10^5$ ， $0 \leq a_i < 2^{23}$ ， $q \in \{1, 2, 3\}$ 。

- 子任务 1 (20 分)： $n \leq 1000$ 。
- 子任务 2 (25 分)： $a_i < 8192$ 。
- 子任务 3 (15 分)： $q = 1$ 。
- 子任务 4 (15 分)： $q = 2$ 。
- 子任务 5 (15 分)： $q = 3$ 。
- 子任务 6 (10 分)：无特殊限制。