

# NOIP模拟赛

| 中文题目名称 | 糗大了    | 快哭了    | 色      | 发怒     |
|--------|--------|--------|--------|--------|
| 题目类型   | 传统     | 传统     | 传统     | 传统     |
| 英文题目名称 | qd     | kk     | se     | fn     |
| 源程序文件名 | qd.cpp | kk.cpp | se.cpp | fn.cpp |
| 输入文件名  | qd.in  | kk.in  | se.in  | fn.in  |
| 输出文件名  | qd.out | kk.out | se.out | fn.out |
| 时间限制   | 1s     | 2.5s   | 1s     | 1s     |
| 内存限制   | 512MB  | 1024MB | 512MB  | 512MB  |
| 子任务数目  | 3      | 4      | 4      | 5      |
| 结果比较方式 | 全文比较   | 全文比较   | 全文比较   | 全文比较   |

## 注意事项

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 文件存储在根目录下即可，需要建立子文件夹。
- C++ 中函数 `main()` 的返回值类型必须是 `int`，返回值必须是 `0`。
- 编译命令为 `-lm -O2 -std=c++11 -w`
- 评测环境为 noi Linux, LemonLime, **输入量大的题目请使用较快的读入方式。**

## T1 糗大了(qd)

### 问题描述

qd 有一张  $n$  个点  $m$  条边的简单无向图，每个点有一个正整数的权值。现在 qd 打算按一个顺序依次删除这  $n$  个点。

qd 定义一个连通块的权值为连通块内所有点的权值的和。他想要知道，每次删除了一个点之后，图中所有连通块权值的最大值。如果图中已经不存在连通块了，则输出 `0`。

### 输入格式

第一行两个正整数  $n, m$ 。

第二行  $n$  个正整数  $a_1, \dots, a_n$ ，表示每个点的权值。

接下来  $m$  行，每行两个正整数  $u, v$ ，表示一条边。

下一行一个  $1 \sim n$  的排列  $p_1, \dots, p_n$ ，表示每次删除的点的编号。

# 输出格式

输出一行  $n$  个整数。第  $i$  个数为删除了点  $p_1, \cdots, p_i$  后的答案。

## 样例1

### 输入样例

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

### 输出样例

```
1 | 9
2 | 8
3 | 4
4 | 2
5 | 1
6 | 0
```

## 样例2

见下发文件 qd2.in 与 qd2.out。

此样例与子任务 1 满足同样的约束条件。

## 数据范围与约束

对于全部数据， $1 \leq n, m, a_i \leq 10^5$ 。

| 子任务编号 | 分值 | $n, m \leq$ | 特殊限制      | 子任务依赖 |
|-------|----|-------------|-----------|-------|
| 1     | 30 | 1000        |           |       |
| 2     | 20 | $10^5$      | $a_i = 1$ |       |
| 3     | 50 | $10^5$      |           | 1, 2  |

# T2 快哭了 (kk)

## 问题描述

kk 有一个  $n \times m$  的非负整数矩阵  $A$  (行列下标都从 0 开始)。

定义  $f$  为一个函数, 其输入为一个  $n \times m$  的矩阵  $A$ , 输出为一个  $n \times m$  的矩阵  $B$ , 满足  $B_{i,j} = A_{i,j} \oplus A_{i+1,j} \oplus A_{i,j+1} \oplus A_{i+1,j+1}$  (其中  $\oplus$  为按位异或), 越界的位置的值为 0。

定义  $f^k(A) = f(f^{k-1}(A))$ , 特别地  $f^0(A) = A$ 。换句话说,  $f^k(A)$  表示对矩阵  $A$  进行  $k$  次  $A \leftarrow f(A)$  操作后的结果。

youl 想要知道,  $f^k(A)$  的第 0 行第 0 列处的值是多少。他有  $q$  次这样的询问, 每次都会指定一个非负整数  $k$  问这个问题的答案。

由于数据很多, 请在下发的样例程序 `kk_sample.cpp` 的基础上进行编程, 其中输入输出和生成数据的部分已经实现好了, 你只需要实现 `init` 和 `query` 函数, `init` 函数是在回答询问前进行的预处理, `query` 函数会传入  $k$ , 需要返回这个询问的答案。

## 输入格式

输入包含一行七个整数  $n, m, q, aw, kw, k_1, k_2$ 。其中  $k_1, k_2$  是随机数生成器的种子,  $aw, kw$  的意义见后文。

## 输出格式

设第  $i$  次询问的答案是  $ans_i$ , 则应输出  $\bigoplus_{i=1}^q i \times ans_i$ 。

## 样例1

### 输入样例

```
1 | 4 3 4 4 2 15109402569541188053 9005878083635208240
```

### 输出样例

```
1 | 42
```

### 样例1解释

矩阵  $A$  以及其进行三次  $A \leftarrow f(A)$  操作后是这样的

$$\begin{bmatrix} 14 & 9 & 8 \\ 4 & 5 & 1 \\ 12 & 14 & 3 \\ 4 & 8 & 0 \end{bmatrix} \xrightarrow{f} \begin{bmatrix} 6 & 5 & 9 \\ 3 & 9 & 2 \\ 14 & 5 & 3 \\ 12 & 8 & 0 \end{bmatrix} \xrightarrow{f} \begin{bmatrix} 9 & 7 & 11 \\ 1 & 13 & 1 \\ 15 & 14 & 3 \\ 4 & 8 & 0 \end{bmatrix} \xrightarrow{f} \begin{bmatrix} 2 & 0 & 10 \\ 13 & 1 & 2 \\ 13 & 5 & 3 \\ 12 & 8 & 0 \end{bmatrix}$$

询问的  $k$  分别是 1, 2, 3, 0, 答案分别是 6, 9, 2, 14。

## 样例2

见下发文件 `kk2.in` 与 `kk2.out`。

此样例与子任务 1 满足同样的约束条件。

# 数据范围与约束

对于全部数据，  
 $1 \leq n \times m \leq 5 \times 10^6, 1 \leq q \leq 5 \times 10^7, 1 \leq aw, kw \leq 31, 0 \leq a_{i,j} < 2^{aw}, 0 \leq k < 2^{kw}, 0 \leq k_1, k_2 < 2^{64}$ 。

| 子任务编号 | 分值 | $n \times m \leq$ | $q \leq$        | $aw \leq$ | $kw \leq$ | 子任务依赖 |
|-------|----|-------------------|-----------------|-----------|-----------|-------|
| 1     | 30 | 1000              | $5 \times 10^7$ | 31        | 10        |       |
| 2     | 10 | $5 \times 10^6$   | $5 \times 10^7$ | 1         | 20        |       |
| 3     | 30 | $5 \times 10^6$   | $5 \times 10^7$ | 31        | 20        | 1, 2  |
| 4     | 30 | $5 \times 10^6$   | $5 \times 10^7$ | 31        | 31        | 3     |

# T3 色 (se)

## 问题描述

se 有一条数轴，上面有  $n$  条线段，第  $i$  条为  $[l_i, r_i]$ 。现在他想把这  $n$  条线段划分为不超过  $k$  个集合，每条线段必须恰好属于其中一个集合。定义一个集合的权值为集合里所有线段的交的长度，定义一种划分方案的权值为所有集合的权值之和。求所有满足条件的划分方案中最大的权值。

## 输入格式

第一行两个正整数  $n, k$ 。  
接下来  $n$  行，每行两个正整数  $l_i, r_i$ ，表示一条线段。

## 输出格式

输出一行一个整数表示最大的权值。

## 样例1

### 输入样例

|   |      |
|---|------|
| 1 | 4 3  |
| 2 | 1 7  |
| 3 | 9 20 |
| 4 | 5 15 |
| 5 | 4 10 |

### 输出样例

|   |    |
|---|----|
| 1 | 24 |
|---|----|

样例解释

分成  $\{[1, 7], [4, 10]\}, \{[9, 20]\}, \{[5, 15]\}$ , 权值为  $3 + 11 + 10 = 24$ 。

样例2

输入样例

|   |    |    |
|---|----|----|
| 1 | 5  | 3  |
| 2 | 15 | 16 |
| 3 | 9  | 14 |
| 4 | 14 | 20 |
| 5 | 4  | 9  |
| 6 | 8  | 14 |

输出样例

|   |    |
|---|----|
| 1 | 12 |
|---|----|

样例解释

分成  $\{[15, 16], [9, 14], [4, 9]\}, \{[14, 20]\}, \{[8, 14]\}$ , 权值为  $0 + 6 + 6 = 12$ 。

样例3

见下发文件 **se3.in** 与 **se3.out**。

此样例与子任务 2 满足同样的约束条件。

数据范围与约束

对于全部数据,  $1 \leq k \leq n \leq 5000, 1 \leq l_i < r_i \leq 10^6$ 。

| 子任务编号 | 分值 | $n \leq$ | 特殊限制    | 子任务依赖 |
|-------|----|----------|---------|-------|
| 1     | 20 | 12       |         |       |
| 2     | 30 | 500      |         | 1     |
| 3     | 10 | 5000     | $k = 2$ |       |
| 4     | 40 | 5000     |         | 2, 3  |

T4 发怒 (fn)

问题描述

fn 有一棵  $n$  个点的树，树上每个点有一个正整数权值  $a_i$ 。定义点集的一个子集  $S$  是连通的，当且仅当在树上仅保留  $S$  内的点以及它们之间的边后，这张图是连通的，定义  $S$  的权值为其包含的所有节点的权值之积。fn 想要知道，这棵树上有多少非空的连通的且权值  $\leq m$  的子集  $S$ ，答案对  $10^9 + 7$  取模。

## 输入格式

第一行两个正整数  $n, m$ 。

第二行  $n$  个正整数  $a_1, \dots, a_n$ ，表示每个点的权值。

接下来  $n - 1$  行，每行两个正整数  $u, v$ ，表示树上的一条边。

## 输出格式

输出一行一个整数，表示答案。

## 样例1

### 输入样例

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

### 输出样例

```
1 | 8
```

### 样例解释

符合条件的点集有：  $\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 4\}$ 。

## 样例2

见下发文件 **fn2.in** 与 **fn2.out**。

此样例与子任务 2 满足同样的约束条件。

## 数据范围与约束

对于全部数据，  $1 \leq n \leq 2000, 1 \leq m \leq 10^6, a_i \leq m$ 。

| 子任务编号 | 分值 | $n \leq$ | $m \leq$ | 特殊限制                   | 子任务依赖   |
|-------|----|----------|----------|------------------------|---------|
| 1     | 5  | 2000     | $10^6$   | $u_i = i, v_i = i + 1$ |         |
| 2     | 10 | 100      | 2000     |                        |         |
| 3     | 20 | 2000     | 2000     |                        | 2       |
| 4     | 20 | 100      | $10^6$   |                        | 2       |
| 5     | 45 | 2000     | $10^6$   |                        | 1, 3, 4 |

