

NOIP 模拟赛

grass8cow

题目名称	签到题	结论题	简单题	套路题
输入文件名	math.in	graph.in	count.in	ds.in
输出文件名	math.out	graph.out	count.out	ds.out
源程序名称	math.cpp	graph.cpp	count.cpp	ds.cpp
时间限制	2s	2s	2s	4s
空间限制	1GB	1GB	1GB	1GB
测试点形式	Subtask	Subtask	Subtask	Subtask

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
4. 选手应将各题的源程序放在选手文件夹内，不要建立子文件夹。
5. 不保证题目按难度排序。

A - 签到题

题目描述

给定正整数 n, k ，请求出最大的正整数 a 使得 $a^k \mid n$ 。

输入格式

包含一行，两个整数代表 n, k 。

输出格式

包含一行，仅一个整数，代表你求出的 a 的值。

样例

样例输入

```
54 3
```

样例输出

```
3
```

数据范围

对于所有数据， $1 \leq n, k \leq 10^{18}$ 。

子任务 1 (20%) : $n \leq 10^6$ 。

子任务 2 (30%) : $n \leq 10^{12}$ 。

子任务 3 (20%) : $k \geq 3$ 。

子任务 4 (30%) : 无特殊限制。

B - 结论题

题目描述

给出一个无向图，每个点有点权 a 和颜色 c ，其中颜色只会有红蓝两种。

可以进行若干次操作，每次可以选择一条边 (u, v) 并进行以下操作：

1. 交换 a_u, a_v 。
2. 若 $c_u = c_v$ ，则将 c_u, c_v 同时改变，即若原来是红色则变成蓝色，反之亦然。

给出初始的点权和颜色，以及目标的点权和颜色。

你需要判断能否通过若干次操作使每个点都达到目标。

输入格式

本题有多组测试数据。输入的第一行包含一个正整数 T ，表示测试数据组数。

接下来依次 T 组测试数据。对于每组测试数据：

第一行两个整数 n, m ，表示给出的图的点数。

接下来 m 行，每行两个整数 u, v ，表示该无向图的一条边。

下一行包含 n 个整数，其中第 i 个整数表示初始的 a_i 。

下一行包含 n 个字符，第 i 个字符代表颜色 c_i ，其中 R 代表红色，B 代表蓝色。

接下来两行以相同方式给出目标状态。

输出格式

对于每组数据，输出一行一个字符串 YES 或者 NO，YES 代表目标状态能达到，NO 代表不能达到。

样例

样例输入

```
3
2 1
1 2
3 4
RR
4 3
BB
3 2
1 2
2 3
1 1 1
RBR
1 1 1
BBB
3 3
1 2
2 3
3 1
1 1 1
RBR
1 1 1
BBB
```

样例输出

```
YES
NO
YES
```

数据范围

对于所有数据, $1 \leq \sum n, \sum m \leq 10^5$, 点权在 1 到 10^6 之间, 不包含自环, 可能有重边, 不保证图连通。

子任务 1 (10%): $n \leq 5$ 。

子任务 2 (20%): 保证给出的图是树。

子任务 3 (20%): 保证给出的图是二分图。

子任务 5 (50%): 无特殊限制。

C - 简单题

题目描述

给出 n, x, y 。令 $f_{n,x,y}(a, b) = \sum_{i=a}^{n-b} \binom{i}{a} x^{i-a} \binom{n-i}{b} y^{n-i-b}$ 。

有 q 次询问，第 i 次询问给出 a_i, b_i ，请你求出 $f_{n,x,y}(a_i, b_i)$ 对 998244353 取模的结果。

输入格式

第一行包含四个整数 n, x, y, q 。

接下来 q 行，每行包含两个整数 a_i, b_i ，表示一次询问。

输出格式

对于每次询问，输出一行一个整数，表示答案对 998244353 取模的结果。

样例

样例输入

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

样例输出

```
6
1
```

数据范围

对于所有数据， $2 \leq n \leq 10^9, 1 \leq x, y < 998244353, 1 \leq q \leq 2 * 10^5, 1 \leq a_i, b_i \leq 5000, a_i + b_i \leq n$ 。

子任务 1 (20%)： $n, q \leq 5 * 10^3$ 。

子任务 2 (10%)： $x = y$ 。

子任务 3 (20%) : $a_i, b_i \leq 20$ 。

子任务 4 (20%) : $a_i, b_i \leq 400$ 。

子任务 5 (30%) : 无特殊限制。

D - 套路题

现在有 10^9 个花盆，依次编号为 $1, 2, \dots, 10^9$ 。

给定 n 个二元组 $L_i, R_i (L_i < R_i)$ ，我们将进行 n 次以下操作：

1. 设当前第 i 个二元组的权值是满足花盆 x 为空且 $L_i < x \leq R_i$ 的 x 数量。选出未被选过且权值最小的二元组 p 。如果存在多个权值最小的二元组，选择其中编号最小的。
2. 对于 $L_p < x \leq R_p$ ，在花盆 x 中种上花。

请求出每次操作选择出的二元组 p 。

保证对于 $1 \leq i < j \leq n, R_i - L_i \leq R_j - L_j$ 。

输入格式

第一行包含一个整数 n ，表示二元组的个数。

接下来 n 行，每行包含两个整数 L_i, R_i 。

输出格式

输出一行 n 个整数，第 i 个整数表示第 i 次操作中选择的二元组。

样例

样例输入

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

样例输出

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

数据范围

对于所有数据, $1 \leq n \leq 2.5 * 10^5$, $0 \leq L_i < R_i \leq 10^9$, 对于 $1 \leq i < j \leq n$, $R_i - L_i \leq R_j - L_j$ 。

子任务 1 (20%) : $n \leq 5 * 10^3$ 。

子任务 2 (20%) : $L_i = 0$ 或 $R_i = 10^9$ 。

子任务 3 (30%) : $n \leq 4 * 10^4$ 。

子任务 4 (30%) : 无特殊限制。