

NOI 2025 模拟赛

题目名称	宿雾若水遥	缠忆君影梦相见	晓月又经宵
题目类型	传统型	传统型	传统型
目录名	mizue	asano	tsuki
可执行文件名	mizue	asano	tsuki
输入文件名	mizue.in	asano.in	tsuki.in
输出文件名	mizue.out	asano.out	tsuki.out
每个测试点时限	3.0 s	1.0 s	6.0 s
内存限制	1024 MiB	1024 MiB	2048 MiB

编译选项

对于 C++ 语言	<code>-static -O2 -std=c++14</code>
-----------	-------------------------------------

注意事项

1. 文件名（包括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`，值必须为 0。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。
4. 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。

宿雾若水遥 (mizue)

【题目描述】

Yumemizuki Mizuki 有一棵 n 个节点的树。她选取了 m 条树上的链，第 i 条链代表树上 u_i 到 v_i 的路径。

设 S_i 表示第 i 条链上的点集，定义 $w(l,r) = |S_l \cup S_{l+1} \cup \cdots \cup S_r|$ ，即第 $l, l+1, \cdots, r$ 条链的点集的并的大小。

有 q 次询问，每次给出 L, R ，你需要输出 $\sum_{L \leq l \leq r \leq R} w(l,r)$ 的值。

【输入格式】

- 从文件 `mizue.in` 中读入数据。
- 第一行三个正整数 n, m, q 。
- 接下来 $n - 1$ 行每行两个正整数 u, v 表示一条树边。
- 接下来 m 行每行两个正整数 u, v 表示一条链。
- 接下来 q 行每行两个正整数 L, R 表示一次询问。

【输出格式】

- 输出到文件 `mizue.out` 中。
- 对于每组询问，输出一行一个非负整数表示答案。

【样例 1 输入】

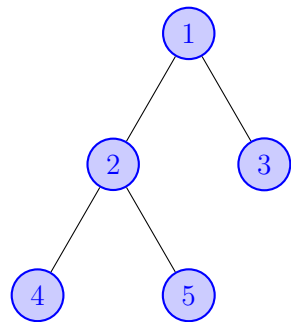
```
1 5 4 4
2 1 2
3 1 3
4 2 4
5 2 5
6 2 4
7 1 5
8 3 4
9 2 5
10 1 3
11 2 4
12 1 4
13 2 3
```

【样例 1 输出】

```
1 23
2 24
3 40
4 12
```

【样例 1 解释】

树的形态：



每条链上的点集： $S_1 = \{2, 4\}$, $S_2 = \{1, 2, 5\}$, $S_3 = \{1, 2, 3, 4\}$, $S_4 = \{2, 5\}$ 。

$w(l, r)$ 的具体值：

- $w(1, 1) = 2, w(1, 2) = 4, w(1, 3) = 5, w(1, 4) = 5$
- $w(2, 2) = 3, w(2, 3) = 5, w(2, 4) = 5$
- $w(3, 3) = 4, w(3, 4) = 5$
- $w(4, 4) = 2$

【样例 2,3,4,5,6】

见选手目录下的 `mizue/mizue2,3,4,5,6.in` 与 `yumemi/yumemi2,3,4,5,6.ans`。
它们分别满足测试点 1, 4 ~ 5, 6 ~ 9, 10 ~ 13, 17 ~ 20 的约束。

【数据范围与提示】

对于所有数据， $1 \leq n, m \leq 2 \times 10^5, 1 \leq q \leq 5 \times 10^5$ 。各测试点的详细约束如下：

测试点编号	$n, m \leq$	$q \leq$	特殊性质
1	10	10	无
2	200	200	
3	1000	1000	
4 ~ 5	2000	10^5	
6 ~ 9	10^5	2×10^5	树边满足 $v = u + 1$
10 ~ 13			第 i 条树边满足 u 在 $[1, i - 1]$ 内随机生成， $v = i$
14 ~ 16			无
17 ~ 20	2×10^5	5×10^5	

缠忆君影梦相见 (asano)

【题目描述】

在梦中，Naganohara Yoimiya 遇到了一个 n 点 m 边的有向图（无重边，无自环）。她发现这张图并不是强连通的，于是她往图中添加了恰好三条边（不是重边和自环）。

从梦中醒来后，Yoimiya 忘掉了原本的图是什么样的，但她记住了添加三条边之后的有向图。

现在她找到了 Yumemizuki Mizuki，她希望 Mizuki 帮她求出最初的不强连通的图有多少种可能。当然，也有可能是 Yoimiya 记错了，也就是说 Yoimiya 记住的这张图删掉任意三条边都仍然是强连通的，这时 Mizuki 只需要回答 0。

认为两张图不同，当且仅当存在一对 u, v 使得第一张图中存在有向边 (u, v) 而第二张图中没有。

【输入格式】

从文件 `asano.in` 中读入数据。

本题有多组数据。第一行一个正整数 T 表示数据组数。对于每组数据：

第一行两个正整数 n, m 。

接下来 m 行，每行两个正整数 u, v 表示一条有向边。

保证无重边（但可以有二元环），无自环。

【输出格式】

输出到文件 `asano.out` 中。

对于每组数据，输出一行一个整数表示可能的最初的图的数量。

【样例 1 输入】

```
1 3
2 3 3
3 1 2
4 2 3
5 3 1
6 3 4
7 1 2
8 2 1
9 1 3
10 2 3
11 4 7
12 1 2
13 2 3
14 3 4
15 4 1
16 1 3
17 2 4
18 3 1
```

【样例 1 输出】

```
1 1
2 4
3 34
```

【样例 1 解释】

对于第一组数据，原图只有一种可能，即三个点，没有任何边的图。

对于第二组数据，由于图本来就不强连通，图中任意去掉三条边后都有可能是原本的图，方案数为 $\binom{4}{3} = 4$ 。

对于第三组数据，一种可能的原本的图是只有 $1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, 1 \rightarrow 3$ 这 4 条边的图；Yoimiya 添加的三条边分别是 $4 \rightarrow 1, 2 \rightarrow 4, 3 \rightarrow 1$ 。

【样例 2,3,4,5】

见选手目录下的 *asano/asano2,3,4,5.in* 与 *asano/asano2,3,4,5.ans*。

样例 2 满足测试点 1 ~ 3 的约束，样例 3 满足测试点 7 ~ 8 的约束，样例 4 满足测试点 9 ~ 10 的限制，样例 5 可能不满足任何特殊性质。

【数据范围与提示】

对于所有数据，保证 $3 \leq n \leq 50, 3 \leq m \leq n(n - 1), 1 \leq T \leq 10$ 。

每个测试点的具体约束如下：

测试点编号	特殊性质
1 ~ 3	$m \leq 80$
4 ~ 6	$m \leq 200$
7 ~ 8	$m \geq (n - 1)^2 + 2$
9 ~ 10	图随机生成
11 ~ 14	$n \leq 40$
15 ~ 20	无

其中随机生成的含义是：先不随机地指定两个正整数 n, m 作为图的点数和边数，再从 $n(n - 1)$ 条所有可能的有向边中随机选取 m 条作为实际的图中的边。

晓月又经宵 (tsuki)

【题目描述】

Yoimiya 正在学习单调栈。她遇到了这样一道例题：给定一个长为 n 的序列 a_0, \dots, a_{n-1} ，求一个子区间 $0 \leq l \leq r \leq n-1$ ，最大化 $\min(a_l, a_{l+1}, \dots, a_r) \times (r-l+1)$ 的值。Yoimiya 很快就解决了这个问题，对于序列 a ，她定义上述问题的答案（即所有子区间 $\min \times \text{len}$ 的最大值）为 $f(a)$ 。

现在 Mizuki 有一个序列 b ，她会吧 a, b 两个序列以任意顺序归并，假设归并后的序列是 c ，她希望最大化 $f(c)$ 的值。这里归并两个序列 a, b 得到 c 的含义是，能够把 c 划分成两个不交的子序列，使得一个是 a 一个是 b 。

这个问题还是太简单了，因此 Mizuki 决定进行 q 次询问，每次询问会给出 $[l, r]$ ，你需要帮她求出：假如现在要把 $b[l \dots r]$ 和 a 进行归并，得到的序列的 f 值最大是多少。

【输入格式】

从文件 **tsuki.in** 中读入数据。

第一行三个正整数 n, m, q 表示序列 a, b 的长度和询问次数。

第二行 n 个正整数 a_0, \dots, a_{n-1} 。

第三行 m 个正整数 b_0, \dots, b_{m-1} 。

接下来 q 行，第 $i+4$ 两个正整数 l_i, r_i ，表示一次询问。

【输出格式】

输出到文件 **tsuki.out** 中。

对于每组询问，输出一行一个正整数表示答案。

【样例 1 输入】

```
1 5 4 2
2 3 3 1 6 1
3 3 5 7 6
4 0 1
5 0 3
```

【样例 1 输出】

```
1 12
2 20
```

【样例 1 解释】

对于第一组询问，一种最优方案是归并为序列 $c = (3, 3, 5, 1, 6, 1)$ ，有 $f(c) = 12$ ，方法是选取子区间 $[0, 3]$ 。

对于第二组询问，一种最优方案是归并为序列 $c = (3, 3, 3, 1, 5, 6, 7, 6, 1)$ ，有 $f(c) = 20$ ，方法是选取子区间 $[4, 7]$ 。

其中，标红的元素来自序列 b ，其余元素来自序列 a 。

【样例 2,3,4,5,6,7,8,9,10】

见选手目录下的 `tsuki/tsuki2,3,4,5,6,7,8,9,10.in` 与 `tsuki/tsuki2,3,4,5,6,7,8,9,10.ans`。

其中，样例 i 满足子任务 $i - 1$ 的限制。

【数据范围与提示】

对于所有数据，有 $1 \leq n, m \leq 150000, 1 \leq q \leq 500000, 1 \leq a_i, b_i \leq 10^9, 0 \leq l \leq r \leq m - 1$ 。

每个子任务的具体约束如下：

- Subtask 1 (10 分)： $n, m, q \leq 3000$ 。
- Subtask 2 (8 分)： $q \leq 300$ 。
- Subtask 3 (20 分)：询问的 $[l, r]$ 满足： $\max(b_{l-1}, b_{r+1}) < \min_{k=l}^r b_k$ 。这里认为 $b_{-1} = b_m = 0$ 。
- Subtask 4 (6 分)： $a_i, b_i \leq 50$ 。
- Subtask 5 (14 分)：所有 a_i 均相同。
- Subtask 6 (11 分)：对于 $1 \leq i \leq m - 1$ ， $b_i \geq b_{i-1}$ 。
- Subtask 7 (13 分)：查询满足 $l = 0$ 。
- Subtask 8 (7 分)：所有查询的 $r - l$ 均相同。
- Subtask 9 (11 分)：无特殊限制。