

NOIP2023 模拟赛

by 广州市铁一中学

题目名称	young	bamboo	follow	heart
题目类型	传统题	传统题	传统题	传统题
目录	young	bamboo	follow	heart
可执行文件名	young	bamboo	follow	heart
输入文件名	young.in	bamboo.in	follow.in	heart.in
输出文件名	young.out	bamboo.out	follow.out	heart.out
时间限制	1.0 秒	1.0 秒	5.0 秒	2.0 秒
内存限制	1024 MiB	1024 MiB	1024 MiB	1024 MiB
是否使用 SPJ	否	否	否	否
子任务数量	4	5	6	5

提交源程序文件名：

对于 C++ 语言	triple.cpp	traversal.cpp	destruction.cpp	calculate.cpp

编译选项：

对于 C++ 语言	-O2 -std=c++14 -static

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。
- C/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。
- 提交的程序代码文件的放置位置请参照具体需求。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。

T1 young

题目描述

笑怂在玩嫁接。

笑怂有两棵以 1 为根的有标号有根树，笑怂想通过对第一棵树操作，把第一棵树变得与第二棵树完全相同。笑怂每次可以选择一个 i 和 j ，满足 $i \neq 1$ ，并且 j 不在 i 的子树里，然后把以 i 为根的子树接到 j 节点的下面。请你求出最少操作次数。

输入格式

第一行一个整数 n 。

接下来 $n - 1$ 行，每行两个整数 u, v 表示第一棵树的边。

接下来 $n - 1$ 行，每行两个整数 u, v 表示第二棵树的边。

输出格式

输出一个数表示最小操作次数。

样例

样例 1

输入

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

输出

1	3
---	---

解释

- 三次操作：
1. 把 3 的子树接到 1 下面。
 2. 把 4 的子树接到 3 下面。
 3. 把 5 的子树接到 1 下面。

样例 2

输入

1	10
2	1 2
3	2 3
4	1 4
5	3 5
6	3 6
7	3 7
8	2 8
9	1 9
10	5 10
11	1 2
12	2 3
13	2 4
14	3 5
15	4 6
16	6 7
17	1 8
18	5 9
19	9 10

输出

1	6
---	---

数据范围

子任务编号	$n \leq$	特殊性质	分数
1	5		20
2	10^3		20
3	10^5	保证第二棵树除了 1 号点的度数都是 1	20
4	10^5		40

对于 100% 的数据， $1 \leq n \leq 10^5$ 。

T2 bamboo

题目描述

笑怂在玩逆序对。

笑怂会随机生成一个长度为 n 的正整数序列 a ，每个 a_i 都在 $[1, m]$ 中随机。但是笑怂有一些癖好，所以有 k 个限制，第 i 个限制是 a_{x_i} 不会随机到 y_i 。

请你帮笑怂求出序列 a 逆序对数的期望，对 998244353 取模。保证至少有一个满足要求的序列。

一个序列的逆序对个数即满足 $i < j, a_i > a_j$ 的 (i, j) 个数。

输入格式

第一行三个整数 n, m, k 。

接下来 k 行，每行两个整数 x_i, y_i 。

输出格式

输出一个数表示逆序对数的期望。

样例

样例 1

输入

1	2	3	2
2	1	2	
3	2	1	

输出

1	748683265
---	-----------

解释

四种可能的序列：

- $[1, 2]$, 0 个逆序对
- $[1, 3]$, 0 个逆序对
- $[3, 2]$, 1 个逆序对
- $[3, 3]$, 0 个逆序对

期望 $\frac{1}{4}$ 个逆序对。

样例 2

输入

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

输出

数据范围

子任务编号	$n \leq$	$m \leq$	$k \leq$	分数
1	5	5	5	10
2	1×10^3	1×10^3	0	20
3	200	200	200	20
4	2×10^3	2×10^3	2×10^3	20
5	10^5	10^5	10^5	30

对于 100% 的数据, $1 \leq n, m \leq 10^5, 0 \leq k \leq 10^5, 1 \leq x_i \leq n, 1 \leq y_i \leq m$ 。
 $\forall i \neq j, (x_i, y_i) \neq (x_j, y_j)$ 。

T3 follow

题目描述

笑怂在玩环。

笑怂有一个 n 个点的环, 对于 $i < n$, 点 i 和 $i + 1$ 相邻, 点 n 和点 1 相邻, 每个点有个颜色 a_i 。笑怂在 q 天中每天都有一个容忍度 k_i , 第 i 天时笑怂想把这个环分成最少的连续段, 满足每一段中任意颜色出现次数不超过 k_i 。请你帮笑怂求出每一天可以划分出的最少连续段个数。注意笑怂不会真的划分, 即每一天都是独立的。

输入格式

第一行两个整数 n, q 。

接下来一行, n 个整数 a_i 。

接下来一行, q 个整数 k_i 。

输出格式

输出 q 行, 每行一个整数表示第 i 天的答案。

样例

样例 1

输入

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

输出

1	3
2	2
3	1
4	1
5	1

解释

$k = 1$ 时：分成 $[1, 1], [2, 3], [4, 5]$ 三段。
 $k = 2$ 时：分成 $[1, 3], [4, 5]$ 两段。
 $k = 3, 4, 5$ 时：分成 $[1, 5]$ 一段。

样例 2

输入

1	10	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
---	----	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

输出

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

数据范围

子任务编号	$n \leq$	$q \leq$	分数
1	100	100	20
2	10^3	10^3	10
3	10^5	1	10
4	10^5	500	20
5	5×10^4	5×10^4	20
6	10^5	10^5	20

对于 100% 的数据, $1 \leq n, q \leq 100000, 1 \leq a_i \leq n, 1 \leq k_i \leq n$ 。

T4 heart

题目描述

笑怂在玩树。

笑怂有一颗 n 个点的有根树，根为 1。给定一个 $1 \sim n$ 的排列 p 。笑怂每次可以走到当前节点的儿子，或者使用传送术。假设笑怂站在节点 i ，则笑怂能传送到节点 j 当且仅当 $i \neq j, p_j < p_i$ ， j 在 i 的子树内，并且对于 i 到 j 的路径上的每个节点 k ，满足 $p_k \geq p_i$ 或 $p_k \leq p_j$ 。

请你求出从每个点 i 出发走到任意一个节点停止的方案数 f_i ，对 998244353 取模。注意，在某一步中使用传送术传送到儿子节点和直接走到儿子节点看作两种不同的方案。

输入格式

第一行一个整数 n 。

接下来 $n - 1$ 行，每行两个整数 u_i, v_i ，表示 u_i, v_i 之间有一条边。

接下来一行 n 个整数，表示 p_i 。

输出格式

输出 n 行 n 个整数 f_i 。

样例

样例 1

输入

1	3
2	2 1
3	3 2
4	1 3 2

输出

1	4
2	3
3	1

解释

从一号点出发的四种路径(－ 代表直接走，－ > 代表传送)：

- 1. 1。
- 2. 1－2。
- 3. 1－2－3。
- 4. 1－2－>3。

样例 2

输入

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

输出

1	8
2	6
3	2
4	1
5	1

数据范围

子任务编号	$n \leq$	特殊性质	分数
1	3×10^3		10
2	10^5	$u_i = i + 1, v_i = i$	30
3	5×10^4		20
4	10^5		10
5	5×10^5		30

对于 100% 的数据, $1 \leq n \leq 5 \times 10^5$ 。