

第四题 置换

提交文件: permutation.cpp
 输入文件: permutation.in
 输出文件: permutation.out
 时间空间限制: 1 秒, 512 MB

Moon 最近在玩一款名为 Shadowverse 的卡牌游戏, 在非常有趣的游戏过程中, Moon 想到这样一个关于洗牌的问题。假设当前牌堆中有 n 张牌, 第 i 张牌的标号为 i , 我们定义一种洗牌方式是一个排列 $X=\{x_1, x_2, \dots, x_n\}$, 也就是把牌堆中第 i 张位置的牌变成第 x_i 张。那么假设现在 Moon 按照 X 的洗牌方式洗了 k 次牌, 不妨设最终得到了一个排列 $Y=\{y_1, y_2, \dots, y_n\}$, y_i 表示洗完牌后第 i 张牌的标号。Moon 希望你可以帮助他算出有多少合法的洗牌方式 X , 满足洗了 K 次后变成排列 Y , 由于答案可能很大, 所以你只需要输出对 998244353 取模的答案即可。

形式化而言, 考虑对于排列 $P=\{p_1, p_2, \dots, p_n\}$ 和排列 $Q=\{q_1, q_2, \dots, q_n\}$, 定义这两个排列的乘积:

$$P \times Q = \{q_{p_1}, q_{p_2}, \dots, q_{p_n}\}$$

而排列 X 的 k 次幂 X^k 为 k 个排列 X 的乘积, 现在考虑给定排列 Y 和正整数 k , 求满足方程 $X^k = Y$ 的排列 X 的数量, 对 998244353 取模。

输入格式

第一行是一个整数 T 表示测试数据组数。

每组数据包括两行, 第一行两个正整数 n, k , 分别表示排列 X 和 Y 的长度、洗了 k 次牌。

第二行是 n 个 1 到 n 内互不相同的正整数 $\{y_1, y_2, \dots, y_n\}$, 表示排列 Y 。

输出格式

T 行, 每行一个整数, 表示合法的洗牌方式的数量, 对 998244353 取模。

样例数据

permutation.in	permutation.out
1 5 6 2 1 4 3 5	2
见/example/permutation/下的 permutation1.in	见/example/permutation/下的 permutation1.out

样例解释

样例中, $X=[3,4,2,1,5]$ 或者 $[4,3,1,2,5]$, 共两个合法排列。

数据范围

对于所有的数据, 有 $1 \leq n \leq 3000, 1 \leq k \leq 10^6, 1 \leq T \leq 10$;

对于 20% 的数据, 有 $1 \leq n, k \leq 8$;
对于另外 10% 的数据, 仅保证 $1 \leq n \leq 8$;
对于另外 30% 的数据, 仅保证 $1 \leq n \leq 50$ 。