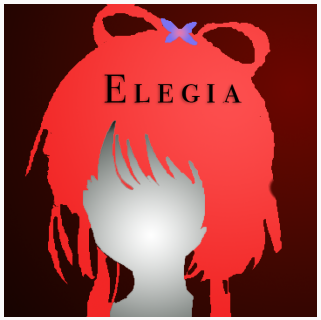
另一个欧拉数问题

Elegia
李白天

2023 年 1 月 16 日

IIIS, THU



題意

题意

对于正整数 α, 考虑下述长为 αn 的序列 a:

- 对于每个 $k = 1, \ldots, n$, 序列 a 中出现了恰好 α 个 k.
- 对于 $i < j$ 满足 $a_i = a_j$, 那么对任意 $i < k < j$, 有 $a_k \geq a_i$.

我们称满足上述要求的序列是一个 n 阶的 α-排列. 现在输入一个 n_0 阶 α-排列 P. 又给定 n, m, 请你计算有多少 n 阶 α-排列包含子序列 P, 并且满足:

- 总共有 m 个下标 i 满足 $a_i > a_{i+1}$.

只需计算出这样的序列总数对 998244353 取模的结果.

算法 1

算法 1 — 人口普查

不难注意到, 对于输入的序列 P, 我们只关心它具有多少个下标满足 $a_i > a_{i+1}$, 记为 m_0.

考虑将所有值为 n 的数插到序列中, 根据要求, 它们此时必须是相邻的, 分类讨论所插入的位置是否对 m 有所贡献, 得到递推式

$$F_{n,m} = (\alpha(n-1)+1-m)F_{n-1,m-1} + (m+1)F_{n-1,m}.$$

初始条件是 $F_{n_0,m_0} = 1$.

算法 1 — 人口普查

不难注意到, 对于输入的序列 P, 我们只关心它具有多少个下标满足 $a_i > a_{i+1}$, 记为 m_0.

考虑将所有值为 n 的数插到序列中, 根据要求, 它们此时必须是相邻的, 分类讨论所插入的位置是否对 m 有所贡献, 得到递推式

$$F_{n,m} = (\alpha(n-1)+1-m)F_{n-1,m-1} + (m+1)F_{n-1,m}.$$

初始条件是 $F_{n_0,m_0} = 1$.

预计得分 10%.

算法 2

算法 2 — 经典 Eulerian 数

$\alpha = 1, n_0 = 1$ 的情况就是经典的 Eulerian 数问题, 有 114514 种推法, 这里给出比较简短有趣的一种.

算法 2 — 经典 Eulerian 数

$\alpha = 1, n_0 = 1$ 的情况就是经典的 Eulerian 数问题, 有 114514 种推法, 这里给出比较简短有趣的一种.

考虑 n 个实数的均匀分布 $(a_1, a_2, \cdots, a_n) \in (0,1)^n$, 根据对称性, a_i 的大小关系服从排列 p_i 是等概率的. 也就是说我们转而考虑 $a_i < a_{i+1}$ 的位置有 k 个的概率.

算法 2 — 经典 Eulerian 数

$\alpha = 1, n_0 = 1$ 的情况就是经典的 Eulerian 数问题, 有 114514 种推法, 这里给出比较简短有趣的一种.

考虑 n 个实数的均匀分布 $(a_1, a_2, \cdots, a_n) \in (0,1)^n$, 根据对称性, a_i 的大小关系服从排列 p_i 是等概率的. 也就是说我们转而考虑 $a_i < a_{i+1}$ 的位置有 k 个的概率.

试想象序列 $\{a_n\}$ 是序列 $\{b_n\}$ 前缀和对 1 取模的结果, 其中 b 也是 $(0,1)$ 上独立均匀随机生成.

算法 2 — 经典 Eulerian 数

那么, $a_i > a_{i+1}$ 的数量就可以通过 $\sum b_n$ 取整的值刻画. 我们希望求 $\sum b_n \in [m, m+1]$ 的概率.

算法 2 — 经典 Eulerian 数

那么, $a_i > a_{i+1}$ 的数量就可以通过 $\sum b_n$ 取整的值刻画. 我们希望求 $\sum b_n \in [m, m+1]$ 的概率.

这是一个高维空间中几何体的体积 (测度). 可以先假设没有 < 1 的限制, 那么总共的测度是 $\frac{(n-k)^n}{n!}$. 进行容斥, 假设有 $j (j \leq n-k)$ 个数强制 ≥ 1, 那么这一部分的测度是 $\frac{(n-k-j)^n}{n!}$.

算法 2 — 经典 Eulerian 数

那么, $a_i > a_{i+1}$ 的数量就可以通过 $\sum b_n$ 取整的值刻画. 我们希望求 $\sum b_n \in [m, m+1]$ 的概率.

这是一个高维空间中几何体的体积 (测度). 可以先假设没有 < 1 的限制, 那么总共的测度是 $\frac{(n-k)^n}{n!}$. 进行容斥, 假设有 $j (j \le n-k)$ 个数强制 ≥ 1, 那么这一部分的测度是 $\frac{(n-k-j)^n}{n!}$.

进行一下整理, 第 n 行的 Eulerian 数的生成函数是

$$(1-x)^{n+1} \left(\sum_{k=0}^{\infty} (k+1)^n x^k \right).$$

这给出 $O(n)$ 计算的方法, 预计得分 10%.

算法 2 — 经典 Eulerian 数

那么, $a_i > a_{i+1}$ 的数量就可以通过 $\sum b_n$ 取整的值刻画. 我们希望求 $\sum b_n \in [m, m+1]$ 的概率.

这是一个高维空间中几何体的体积 (测度). 可以先假设没有 < 1 的限制, 那么总共的测度是 $\frac{(n-k)^n}{n!}$. 进行容斥, 假设有 $j (j \le n-k)$ 个数强制 ≥ 1, 那么这一部分的测度是 $\frac{(n-k-j)^n}{n!}$.

进行一下整理, 第 n 行的 Eulerian 数的生成函数是

$$(1-x)^{n+1} \left(\sum_{k=0}^{\infty} (k+1)^n x^k \right).$$

这给出 $O(n)$ 计算的方法, 预计得分 10%.

进一步能得到二元生成函数

$$A(x,t) = \frac{(1-t)e^{x(1-t)}}{1-te^{x(1-t)}}.$$

算法 3

算法 3

对于 $\alpha = 1$ 的情况, 可以从组合入手.

算法 3

对于 $\alpha = 1$ 的情况, 可以从组合入手.

我们现在有 m_0 个 $>$, 那么接下来插入的 $n - n_0$ 个数插在已有的 $n_0 + 1$ 个空隙中. 每个空隙的贡献都是欧拉数加上原有的贡献, 除了最靠右的空隙要单独讨论.

算法 3

对于 $\alpha = 1$ 的情况, 可以从组合入手.

我们现在有 m_0 个 $>$, 那么接下来插入的 $n - n_0$ 个数插在已有的 $n_0 + 1$ 个空隙中. 每个空隙的贡献都是欧拉数加上原有的贡献, 除了最靠右的空隙要单独讨论.

通过二元生成函数来做可以直接计算出来.

算法 3

对于 $\alpha = 1$ 的情况, 可以从组合入手.

我们现在有 m_0 个 $>$, 那么接下来插入的 $n - n_0$ 个数插在已有的 $n_0 + 1$ 个空隙中. 每个空隙的贡献都是欧拉数加上原有的贡献, 除了最靠右的空隙要单独讨论.

通过二元生成函数来做可以直接计算出来.

化简后得到的形式是

$$(1-x)^{n+1}\left(\sum_{k=0}^{\infty}(k+1)^{n-n_0}\binom{k-m_0+n_0}{n_0}x^k\right).$$

可以 $O(n)$ 提取一项系数, 加上上个子任务预计得分 40%.

这个形式看起来具有美丽的结构.

$$(1-x)^{n+1}\left(\sum_{k=0}^{\infty}(k+1)^{n-n_0}\binom{k-m_0+n_0}{n_0}x^k\right).$$

这个形式看起来具有美丽的结构.

$$(1-x)^{n+1}\left(\sum_{k=0}^{\infty}(k+1)^{n-n_0}\binom{k-m_0+n_0}{n_0}x^k\right).$$

考虑我们最开始用来打暴力的递推系统, 在 $\alpha=1$ 的情况下是

$$E_{n,m}=(n-m)E_{n-1,m-1}+(m+1)E_{n-1,m}.$$

这个形式看起来具有美丽的结构.

$$(1-x)^{n+1}\left(\sum_{k=0}^{\infty}(k+1)^{n-n_0}\binom{k-m_0+n_0}{n_0}x^k\right).$$

考虑我们最开始用来打暴力的递推系统, 在 $\alpha=1$ 的情况下是

$$E_{n,m}=(n-m)E_{n-1,m-1}+(m+1)E_{n-1,m}.$$

如果我们填下了第 n_0 行, 如何计算出递推得到的 n 行?

这个形式看起来具有美丽的结构.

$$(1-x)^{n+1}\left(\sum_{k=0}^{\infty}(k+1)^{n-n_0}\binom{k-m_0+n_0}{n_0}x^k\right).$$

考虑我们最开始用来打暴力的递推系统, 在 $\alpha=1$ 的情况下是

$$E_{n,m}=(n-m)E_{n-1,m-1}+(m+1)E_{n-1,m}.$$

如果我们填下了第 n_0 行, 如何计算出递推得到的 n 行? 根据前述答案的线性性, 我们应该这么做:

$$\frac{E_{n+1}}{(1-x)^{n+1}}=\left(\frac{d}{dx}x\right)^{n-n_0}\left(\frac{E_{n_0}(x)}{(1-x)^{n_0+1}}\right).$$

令 $G_n(x) = xF_n(x)/(1-x)^{\alpha n+1}$ 时, 我们将第 n 行到第 $n+1$ 行的递推进行了变换

$$\begin{array}{ccc} F_n(x) & \xrightarrow{\quad T_n \quad} & F_{n+1}(x) \\ \downarrow & & \downarrow \\ G_n(x) & \xrightarrow{\frac{x}{(1-x)^{\alpha-1}}\frac{d}{dx}} & G_{n+1}(x) \end{array}$$

注意下面的变换是**和 n 无关**的.

算法 4

算法 4

先将 F 转换到 G 下进行计算, 然后还原回 F.

算法 4

先将 F 转换到 G 下进行计算, 然后还原回 F.

当 $\alpha = 2$ (一般地, α 为小常数) 的时候, 变换写为

$$G_n = \frac{x}{1-x} G'_{n-1},$$

注意对应到数组上的递推式是

$$G_{n,m} = mG_{n-1,m} + G_{n,m-1}$$

算法 4

先将 F 转换到 G 下进行计算, 然后还原回 F.

当 $\alpha = 2$ (一般地, α 为小常数) 的时候, 变换写为

$$G_n = \frac{x}{1-x} G'_{n-1},$$

注意对应到数组上的递推式是

$$G_{n,m} = mG_{n-1,m} + G_{n,m-1}$$

考虑转置 (对列建生成函数), 发现有

$$G^{\mathsf{T}}_m = \frac{c_m x^{n_0} + G^{\mathsf{T}}_{m-1}}{1-mx},$$

其中 c_m 是修正项, 用于计算递推起点的影响.

算法 4 — 转置原理

这是转置原理的典型应用, 我们可以得到一个算法在 $O(n \log^2 n)$ 时间内进行递推.

算法 4 — 转置原理

这是转置原理的典型应用，我们可以得到一个算法在 $O(n \log^2 n)$ 时间内进行递推.

不过对本题而言，由于我们只询问一个数，做法的常数会小一点. 答案只关乎 G_m^{T} 的一个线性组合在 n 次项的系数，我们只想求

$$\sum_k b_k G_k^{\mathsf{T}} = \sum_{l \le r} \frac{c_l b_r}{(1 - lx) \cdots (1 - rx)}.$$

算法 4 — 转置原理

这是转置原理的典型应用，我们可以得到一个算法在 $O(n \log^2 n)$ 时间内进行递推.

不过对本题而言，由于我们只询问一个数，做法的常数会小一点. 答案只关乎 G_m^{T} 的一个线性组合在 n 次项的系数，我们只想求

$$\sum_k b_k G_k^{\mathsf{T}} = \sum_{l \le r} \frac{c_l b_r}{(1 - lx) \cdots (1 - rx)}.$$

时间复杂度 $O(n \log^2 n)$，预计得分 30%.

算法 5

算法 5 — 基的变换

比较

$$E_n = xE'_{n-1}, \quad G_n = \frac{x}{(1-x)^{\alpha-1}} G'_{n-1}.$$

我们不喜欢后者, 但喜欢前者, 因为前者的变换做 k 次, 只需要个快速幂.

算法 5 — 基的变换

比较

$$E_n = xE'_{n-1}, \quad G_n = \frac{x}{(1-x)^{\alpha-1}} G'_{n-1}.$$

我们不喜欢后者, 但喜欢前者, 因为前者的变换做 k 次, 只需要个快速幂.

让后者变成前者.

算法 5 — 基的变换

比较

$$E_n = xE'_{n-1}, \quad G_n = \frac{x}{(1-x)^{\alpha-1}} G'_{n-1}.$$

我们不喜欢后者, 但喜欢前者, 因为前者的变换做 k 次, 只需要个快速幂.

让后者变成前者.

直接设 $G_n = H_n \circ y(x)$, 我们希望 $H_n(y) = yH'_{n-1}(y)$.

算法 5 — 基的变换

比较

$$E_n = xE'_{n-1}, \quad G_n = \frac{x}{(1-x)^{\alpha-1}} G'_{n-1}.$$

我们不喜欢后者, 但喜欢前者, 因为前者的变换做 k 次, 只需要个快速幂.

让后者变成前者.

直接设 $G_n = H_n \circ y(x)$, 我们希望 $H_n(y) = yH'_{n-1}(y)$.

方程会回应我们的希望, y 只需要是

$$\frac{x}{(1-x)^{\alpha-1}} y'$$

的解.

算法 5 — 基的变换

y 的复合逆是好求的 (但常数上是大头), 因此我们可以容易地算出 $H_{n_0}(y)$ 的关于 y 的系数. 然后算出 $H_n(y)$, 然后 Lagrange 反演公式计算 $F_n(x)$ 的某一项的系数.

算法 5 — 基的变换

y 的复合逆是好求的 (但常数上是大头), 因此我们可以容易地算出 $H_{n_0}(y)$ 的关于 y 的系数. 然后算出 $H_n(y)$, 然后 Lagrange 反演公式计算 $F_n(x)$ 的某一项的系数.

时间复杂度 $O(n \log n)$ (虽然比前面的 $O(n \log^2 n)$ 还慢). 预计得分 100%.

谢谢大家!