4　命运歧途 (fate)

对于每个 k，询问有多少 n 阶排列满足 $|p_i - p_{i+1}| \neq k$。

4.1　每次 $O(n^2)$ DP

我们先考虑 $k = 1$ 怎么做。$f(i, j, 0/1)$ 表示前 i 个数总共有 j 个相邻的数位置现在还是相同的，$0/1$ 表示 i 和 $i-1$ 是不是相邻的，接下来我们考虑 $i+1$ 插在这个排列的哪个位置，只有 $O(1)$ 种转移。

注意到对于 $\bmod k$ 同余的类互不干扰，我们可以按照 $1, k+1, \ldots$，然后 $2, k+2, \ldots$ 的顺序插入排列，可以做任意 k 的 DP。

4.2　容斥原理

我们先考虑 $k = 1$ 怎么做。我们现在有 $n-1$ 个条件：对于第 i 个条件，我们记命题 P_i 是 “i 的位置和 $i+1$ 的位置相邻”。我们要计数 $P_1, \ldots, P_{n-1}$ 均不成立的情况。容斥原理就是展开式子 $(1-P_1)(1-P_2)\cdots(1-P_{n-1})$，那么我们看看一些 P 强制成立的时候会发生什么：

如果 $P_L, P_{L+1}, \ldots, P_R$ 强制成立，那么说明 $|p_L - p_{L+1}| = 1, |p_{L+1} - p_{L+2}| = 1, \ldots, |p_R - p_{R+1}| = 1$，也就是说 $p_L, \ldots, p_{R+1}$ 必然是连续上升或者连续下降的。

因此我们可以考虑这样一个状态：$f_{n,k}$ 表示将一个序列分割为 k 段，设 u 为方案中长度 > 1 的段数，全体 2^u 求和。分割为了 k 段，就说明有 $n-k$ 个要求被强制成立了，因此答案就是

$$\sum_k k!(-1)^{n-k} f_{n,k}$$

注意我们可以通过前缀和优化在 $\Theta(n^2)$ 时间内算出 f 的所有。

4.3　扫动 DP $\Theta(n^{2.5})$

对于一般的 k 的情况，我们是要对 $\bmod k$ 相同的分为一组进行容斥。有 a 个大小为 $\lfloor n/k \rfloor$ 的组和 b 个大小为 $\lceil n/k \rceil$ 的组。我们令 $F_m(x) = \sum_k (-1)^{m-k} f_{m,k} x^k$，那么令 $G(x) = F_{\lfloor n/k \rfloor}(x)^a F_{\lceil n/k \rceil}(x)^b$，答案就是

$$\sum_k k! [x^k] G(x)$$

然而本题采用了任意模数，也是为了提示标算并非 FFT。我们考虑这样一个计算过程：

令 k 从 1 至 n 枚举，我们发现 $G(x)$ 的幂的表示会有重叠的部分，比如 $k \geq n/2$ 的时候有 $G(x) = F_1(x)^{n-2k} F_2(x)^k$。因此我们只需维护一下 $G(x)$ 的改变即可。注意添加/去掉一项 m 次多项式只需要 $\Theta(nm)$ 的时间。总共出现的多项式次数均为 n/k 的上下取整情况，每个相同次数的多项式出现次数乘以多项式本身的次数显然不超过 n，所以

$$\sum \deg F = \Theta(n^{1.5})$$

因此这一做法的复杂度是 $\Theta(n^{2.5})$。

4.4　$\Theta(n^2 \log n)$ 做法

事实上存在一个 $\Theta(n^2 \log n)$ 的无 FFT 做法。

现在我们需要计算 $H(x) = F(x)^a G(x)^b$，考虑求导，有

$$\begin{aligned} H'(x) &= aF(x)^{a-1} G(x)^b F'(x) + bF(x)^a G(x)^{b-1} G'(x) \\ &= H(x) \left(a \frac{F'(x)}{F(x)} + b \frac{G'(x)}{G(x)} \right) \end{aligned}$$

我们按照 $[x^n] H(x) \to [x^n] H(x) F'(x) \to [x^n] \frac{H(x) F'(x)}{F(x)}$ 的顺序，每一步都是 $\Theta(\deg F)$ 的计算，因此这样就可以在 $\Theta(n(\deg F + \deg G))$ 的时间完成快速幂。

这样根据调和级数可知总复杂度就是 $\Theta(n^2 \log n)$ 了，但是由于是任意模数，我们并不能直接根据这个式子做，因为有除法。但所幸我们最终只需要知道所有 $k![x^k]G(x)$。

接下来我们需要将模数拆分成两部分，令 $M = 2^t M'$，其中 M' 是奇数。

4.4.1　模 2^t

设 k 是最小的满足 $2^t \mid k!$，因此我们只需要暴力做快速幂算前 k 项就可以了。注意到 $k = \Theta(t) = O(\log M)$，所以复杂度 $\Theta(n \log^2 M \log n)$。

4.4.2　模 M'

我们接下来考虑一个重要性质：

$F_m(x)$ 不存在常数项，且 $|[x^1]F_m(x)| \le 2$。(特别地，$F_1(x) = x$，而对于 $m \ge 2$ 来说有 $[x^1]F_m(x) = 2(-1)^{m-1}$)。

因为 M' 是奇数，所以就算是 2 我们也可以把它除掉了。我们考虑补充阶乘 $\widehat{F}(x) = \sum_k k! x^k [x^k] F(x)$ 和二项乘法 $\widehat{F} \star \widehat{G} = \widehat{FG}$，我们发现这样 $\widehat{F'}$ 与 $\widehat{F}$ 的关系就消去了除法，等价于一个位移了。

通过这样的转化，我们可以无障碍地计算出 $\widehat{F^a G^b}$，而这足够求出答案。

最后我们再用 CRT 合并一下就得到了最终的答案。时间复杂度 $\Theta(n^2 \log n)$。

由于评测机是 32 位的，所以没有 `__int128` 可以用，Barret Reduction 可能不太方便，但可以使用 Montgomery Reduction 来做取模优化，由于 M' 是奇数，在这里使用 Montgomery Reduction 是很方便的。可参看 Min25 Blog 以及 std 的实现。