## 约定

$F[X]$：域 $F$ 上的一元多项式环。
$F[\![X]\!]$：域 $F$ 上的一元形式幂级数环。
$F(X)$：域 $F$ 上的一元有理分式域。
$F(\!(X)\!)$：域 $F$ 上的一元形式 Laurent 级数环。
$\langle A\rangle$：$A$ 生成的理想。
朴素做法能在 $O(n)$ 时间内实现 $F[X]$ 中的加法，$O(n^2)$ 时间内实现乘法与带余除法。

# 1 有限域

设 $F$ 为域。记 $1_F$ 或 $1$ 为其幺元，$0_F$ 或 $0$ 为其零元。

**定义 (域的特征)** 不难验证 $\mathbb{Z}\to F: a\mapsto a\cdot 1_F$ 为环同态，且其像同构于某个 $\mathbb{Z}/p\mathbb{Z}$。域 $F$ 中没有非零的零因子，于是 $p=0$ 或者 $p$ 为素数。记 $\mathrm{char}(F)$ 为该 $p$，称作域 $F$ 的**特征**。

最简单的有限域是 $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$，其中 $p$ 为素数。$\mathbb{F}_p$ 的特征是 $p$。

## 1.1 域扩张

设 $E, F$ 为域，给定嵌入 $u: F\hookrightarrow E$。我们可将 $F$ 与 $u(F)\subset E$ 等同，在这个意义下，$F$ 可视为 $E$ 的子域。称 $E$ 为 $F$ 的**扩张 (扩域)**，记作 $E|F$。

在域扩张 $E|F$ 中，$E$ 总是一个 $F$-线性空间。若 $E$ 的一个 $F$-子代数本身是域，则称作 $E|F$ 的**子扩张**。

域 $F$ 的最小子域或者是 $\mathbb{Q}$，或者是 $\mathbb{F}_p$，其中 $p$ 是 $F$ 的特征。

**定义 1.1.1** 令 $E|F$ 为域扩张。

- 定义 $E|F$ 的次数为 $[E:F] := \dim_F E$。
- 若 $[E:F]$ 有限，则称 $E|F$ 是**有限扩张**。
- 若每个 $x\in E$ 在 $F$ 上都是代数元，则称 $E|F$ 为**代数扩张**。
- 定义 $F[x]$ 为 $x$ 生成的子代数，$F(x)|F$ 为 $x$ 生成的子扩张。二者分别对应 $F[X]$ 与 $F(X)$。
- 若 $E$ 中有一族元素 $\{x_i\}_{i\in I}$ 满足 $F(x_i: i\in I)=E$，则称 $F$ 为**有限生成扩张**，$\{x_i\}_{i\in I}$ 称作 $E|F$ 的生成集。

**注记** 有限扩张 $E|F$ 必然有限生成。若 $\{x_j\}_1^k$ 为 $F$-线性空间 $E$ 的基，则 $E=F(x_1,\cdots,x_k)$。

## 1.2 极小多项式

设域扩张 $E|F$。对于 $x\in E$，定义其在 $F$ 上的**极小多项式** $P_x\in F[X]$ 为次数最低的使得 $P_x(x)=0$ 的首一多项式。$x$ 的其他所有零化多项式 $Q\in F[X]$ 皆满足 $P_x\mid Q$。

由环论的结论可以得到 $P_x$ 不可约，于是有 $F[x]\xrightarrow{\sim}F[X]/\langle P_x\rangle\xrightarrow{\sim}F(x)$，即 $F[x]$ 是域且 $F[x]=F(x)$。具体地

$$\begin{aligned}F[X]/\langle P_x\rangle &\xrightarrow{\sim} F(x)\\ f+\langle P_x\rangle &\longmapsto f(x):=\mathrm{ev}_x(f)\\ X+\langle P_x\rangle &\longmapsto x,\end{aligned}$$

其中 $\mathrm{ev}_x$ 为取值同态。

对于 $E|F$ 的任意子扩张 $E'|F$，若 $x\in E$ 在 $F$ 上代数，则亦在 $E'$ 上代数，且 $x$ 在 $E'$ 上的极小多项式必整除 $P_x$，因此 $[E'(x):E']\le[F(x):F]$。

**命题 1.2.1 (L. Kronecker)** 设 $P\in F[X]$ 为不可约多项式，定义 $F$-代数 $E:=F[X]/\langle P\rangle$，则 $E|F$ 是域扩张，$[E:F]=\deg P$，且陪集 $x:=X+\langle P\rangle\in E$ 满足 $P(x)=0$。

**例 1.2.2** 复数域 $\mathbb{C}$ 可看作是 $\mathbb{R}[X]/\langle X^2+1\rangle$。$\mathbb{C}|\mathbb{R}$ 是域扩张，$\dim_{\mathbb{R}}\mathbb{C}=2$，且 $i:=X+\langle X^2+1\rangle\in\mathbb{R}[X]/\langle X^2+1\rangle$ 满足 $i^2+1=0$。

## 1.3 有限域的构造

任何有限域 $F$ 都有非零的素数特征（否则导致 $\mathbb{Q}$ 嵌入 $F$）。为进一步说明有限域的结构，我们不加证明地给出如下定理。

**定理 1.3.1** 任何特征为 $p$ 的有限域都是 $\mathbb{F}_p$ 的有限扩张，且 $|F|=p^{[F:\mathbb{F}_p]}$。反过来，对任意 $q=p^m\,(m\ge1)$，在同构意义下都存在唯一的有限域 $F$ 使得 $|F|=q$，记作 $\mathbb{F}_q$，其元素皆满足 $x^q=x$。

**注记** 以上定理给出 $x^p\equiv x\pmod p$，且若 $x\perp p$ 则 $x^{p-1}\equiv1\pmod p$。这正是费马小定理。

让我们先来看一个例子。我们已知 $\mathbb{F}_2$ 是含有 2 个元素的域，我们现在来看怎么从它开始构造出一个含有 4 个元素的域。在 $\mathbb{F}_2[X]$ 上取一个二次不可约多项式，譬如 $X^2+X+1$。于是 $\mathbb{F}_2[X]/\langle X^2+X+1\rangle$ 是域。对于 $f\in\mathbb{F}_2[X]$，我们在 $\mathbb{F}_2[X]$ 中做带余除法

$$f(X)=h(X)(X^2+X+1)+r(X),$$

记 $r(X)=c_0+c_1X$，则

$$\begin{aligned}f(x)+\langle X^2+X+1\rangle &= h(X)(X^2+X+1)+r(X)+\langle X^2+X+1\rangle\\ &= c_0+c_1X+\langle X^2+X+1\rangle\\ &= \big(c_0+\langle X^2+X+1\rangle\big)+\big(c_1+\langle X^2+X+1\rangle\big)\big(X+\langle X^2+X+1\rangle\big).\end{aligned}$$

由于 $c_0, c_1 \in \mathbb{F}_2$，它们只能取 0 或 1。于是 $f(X) + \langle X^2 + X + 1 \rangle$ 必然是以下四个之一：

$$\begin{aligned} 0 &:= 0 + \langle X^2 + X + 1 \rangle, \\ 1 &:= 1 + \langle X^2 + X + 1 \rangle, \\ x &:= X + \langle X^2 + X + 1 \rangle, \\ 1 + x &:= 1 + X + \langle X^2 + X + 1 \rangle. \end{aligned}$$

从而 $\mathbb{F}_2[X]/\langle X^2 + X + 1 \rangle$ 恰有四个元素，分别是 $0, 1, x, 1 + x$。这与我们上面的命题一致。

从这个例子中亦可看出，构造有限域的关键是找到 $\mathbb{F}_p$ 上的 $n$ 元的不可约多项式。

# 2 有限域的算法

我们知道，任何元素个数为 $q$ 的有限域 $\mathbb{F}_q$ 都满足 $q = p^m$，其中 $p$ 为某个质数。这表明，若设 $P(X) \in \mathbb{F}_p[X]$ 为某个 $m$ 次不可约多项式 $P_m$，则 $\mathbb{F}_q \xrightarrow{\sim} \mathbb{F}_p[X]/\langle P(X) \rangle$。因此，我们总可以将有限域 $\mathbb{F}_q$ 中的元素看作是一个小于 $m$ 次的 $\mathbb{F}_p$ 上的多项式。

## 2.1 寻找不可约多项式

设 $q = p^n$，其中 $p$ 为素数。我们的任务是找到 $\mathbb{F}_p[X]$ 上的 $n$ 次首一不可约多项式。

朴素的做法是枚举首一的所有 $p^{n-1}$ 个多项式 $f$，对于每个多项式，枚举次数小于等于 $n/2$ 的所有首一多项式 $g$，逐个判断是否有 $g \mid f$。其时间复杂度为 $O(q\sqrt{q}\log q)$。

若使用 Eratosthenes 筛或线性筛法，则可在更低时间按复杂度内一次性筛出所有 $m\,(m \leq n)$ 次首一不可约多项式。然而，这个优化仍然不是最显著的。

**命题 2.1.1 (C. F. Gauss)** 对任意 $n \geq 1$，有限域 $\mathbb{F}_q$ 上的 $n$ 次首一不可约多项式个数 $\Psi_n(q)$ 有以下公式

$$\Psi_n(q) = \frac{1}{n} \sum_{d \mid n} \mu\Big(\frac{n}{d}\Big) q^d.$$

由此可知 $n\Psi_n(q) = q^n + o(q^n)$。这就说明，任意一个 $n$ 次首一多项式不可约的概率是 $1/n$。根据这点，我们随机选取首一多项式，判断它是否不可约。该做法的时间复杂度为 $O(\sqrt{q}\log^3 q)$。

## 2.2 基本运算

设 $q = p^m$，我们用 $\tilde{O}(\cdot)$ 表忽略 $\log m$ 及 $\log\log q$ 因子后的时间复杂度。基于朴素的多项式运算，我们有

- $\mathbb{F}_q$ 中的加减法能在 $O(\log q)$ 时间内完成。
- $\mathbb{F}_q$ 中的乘法、取模能在 $O(\log^2 q)$ 时间内完成。

其写法无非是一般的多项式运算加上对不可约多项式 $P_m$ 取模。

## 2.3 特征为 2 的有限域

域 $\mathbb{F}_2 := \mathbb{Z}/2\mathbb{Z}$ 是特殊的。观察到，$\mathbb{F}_2$ 仅有 $0,1$ 两个元素，其加法无非是异或运算 xor。这导致特征为 2 的有限域 $\mathbb{F}_{2^m}$ 在计算机中能更方便且快速地进行运算。

这里给出 $\mathbb{F}_2[X]$ 中前一些次数的不可约多项式：

$$
\begin{aligned}
&0,\\
&(11)_2,\\
&(111)_2,\\
&(1011)_2,\\
&(10011)_2,\\
&(100101)_2,\\
&(1011011)_2,\\
&(10000011)_2,\\
&(100011101)_2,\\
&(1000010001)_2,\\
&(10001101111)_2,\\
&(100000000101)_2,\\
&(1000011101011)_2,\\
&(10000000011011)_2,\\
&(100000010101001)_2,\\
&(1000000000110101)_2.
\end{aligned}
$$

我们将这些不可约多项式记到数组 ir[] 中。

```
namespace FFCalc {

    using Felement = unsigned int;
    array<Felement, 16> ir = {...};

    Felement add(Felement a, Felement b) { return a ^ b; }

    Felement reduce(Felement a, size_t n) {
        if (a < (1 << n)) return a;
        return reduce(a ^ (ir[n] << (__lg(a) - n)), n);
    }

    Felement mul(Felement a, Felement b, size_t n) {
        Felement mult = 0;
        for (int i = 0; i < n; i++)
```

```
        if (b & (1 << i)) mult ^= (a << i);
        return reduce(mult, n);
    }

}; // namespace FFCalc
```

# 3 快速算法

上一节给出的做法是最朴素的，我们还可以做得更快，其切入点在于加速多项式的运算。

## 3.1 循环卷积

熟知的公式

$$\widehat{f}_k = \sum_{j=0}^{n-1} f_j \exp\left(-\frac{2k\pi i}{n} j\right) \iff f_m = \frac{1}{n}\sum_{l=0}^{n-1} \widehat{f}_l \exp\left(\frac{2m\pi i}{n} l\right)$$

由矩阵等式

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{pmatrix} = n \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{pmatrix}^{-1}$$

给出，其中 $\omega = \exp(2\pi i/n)$。我们考虑域 $\mathbb{C}[X]/\langle X^n - 1\rangle$，记 $f(X) = \sum_0^{n-1} f_j X^j$ 以及

$$\widehat{f} := \left(\widehat{f}_0, \widehat{f}_1, \cdots, \widehat{f}_{n-1}\right) \in \mathbb{C}^n,$$

则映射 $\mathcal{F} := \left[f \mapsto \widehat{f}\right]$ 是 $\mathbb{C}[X]/\langle X^n - 1\rangle \to \mathcal{C}^n$ 的环同态，这是因为对每个 $k\,(k < n)$ 都有

$$\left(\sum_{j=0}^{n-1} f_j \omega^{kj}\right)\left(\sum_{j=0}^{n-1} g_j \omega^{kj}\right) = \sum_{l=0}^{n-1}\left(\sum_{j=0}^{l} f_j g_{l-j} + \sum_{j=0}^{l+n-1} f_j g_{l+n-1-j}\right)\omega^{kl}.$$

设 $F$ 是域，我们称 $F[X]/\langle X^n - 1\rangle$ 中的乘法称作**循环卷积**。设 $f, g \in \mathbb{C}[X]/\langle X^n - 1\rangle$，则其循环卷积可通过 $\mathcal{F}^{-1}(\mathcal{F}(f)\mathcal{F}(g))$ 计算。事实上，由 DFT 的公式易知 $\mathcal{F}$ 是环同构。

设 $f, g \in F[X]$，令 $n = \deg f + \deg g + 1$，则 $fg$ 在 $F[X]$ 中与 $F[X]/\langle X^n - 1\rangle$ 中的表达式是一样的。因此，取这样的 $n$，则上述离散傅里叶变换亦可用于计算正常的多项式乘法。

## 3.2 快速傅里叶变换 (FFT)

### 3.2.1 分治法

设 $f(X) = \sum_1^{n-1} c_j X^j \in \mathbb{C}[X]/\langle X^n - 1\rangle$。令

$$g(X) = \sum_{2j \le n} c_{2j} X^j, \qquad h(X) = \sum_{2j+1 \le n} c_{2j+1} X^j,$$

则有 $f(X)=g(X^2)+Xh(X^2)$。记 $\omega_n=\exp(2\pi i/n)$，代入 $\omega_n^k$ 与 $\omega_n^{k+n/2}$。显然 $g,h$ 在这两点的取值相同。我们有

$$
\begin{aligned}
f\big((\omega_n^k)^2\big) &= g\big((\omega_n^k)^2\big)+\omega_n^k h\big((\omega_n^k)^2\big)=g\big(\omega_{n/2}^k\big)+\omega_n^k h\big(\omega_{n/2}^k\big),\\
f\big((\omega_n^{k+n/2})^2\big) &= g\big(\omega_n^{2k+n}\big)+\omega_n^{k+n/2} h\big(\omega_n^{2k+n}\big)=g\big(\omega_{n/2}^k\big)-\omega_n^k h\big(\omega_{n/2}^k\big).
\end{aligned}
$$

若能求出 $g,h$ 在 $\omega_{n/2}^k$ 的取值则可以得到 $f$ 在 $\omega_n^k$ 与 $\omega_n^{k+n/2}$ 处的值。注意到前者对应 $\mathcal{C}[X]/\langle X^{n/2}-1\rangle$，于是我们将问题转化为了两个 $n/2$ 时的相同问题。

当 $n=2^m\ (m\in\mathbb{N})$ 时，上述过程可以一直进行下去。这就给出了一个分治的做法。

### 3.2.2 FDFT 的递归法实现

```
using Comp = std::complex<double>;  // STL complex
constexpr Comp I(0, 1);  // i
constexpr int MAX_N = 1 << 20;
Comp tmp[MAX_N];

// rev=1,DFT; rev=-1,IDFT
void DFT(Comp* f, int n, int rev) {
    if (n == 1) return;
    for (int i = 0; i < n; i++) tmp[i] = f[i];

    for (int i = 0; i < n; i++) {
        if (i & 1)
        f[n / 2 + i / 2] = tmp[i];
        else
        f[i / 2] = tmp[i];
    }
    Comp *g = f, *h = f + n / 2;
    DFT(g, n / 2, rev), DFT(h, n / 2, rev);
    Comp cur(1, 0), step(cos(2 * M_PI / n), sin(2 * M_PI * rev /
       n));

    for (int k = 0; k < n / 2; k++) {
        tmp[k] = g[k] + cur * h[k];
        tmp[k + n / 2] = g[k] - cur * h[k];
        cur *= step;
    }
    for (int i = 0; i < n; i++) f[i] = tmp[i];
}
```

#### 3.2.3 位逆序置换

使用位逆序置换 (bit-reversal permutation) 可求出第 $k$ 项在分治到底后的位置。据此，我们不必递归，而是变换位置后直接进行计算。

### 3.3 快速数论变换 (NTT)

以 $R[M]$ 表幺半群环。

DFT 的依据是存在环同构 $\mathbb{C}[X]/\langle X^n-1\rangle \xrightarrow{\sim} \mathbb{C}^n$。现在我们考虑讲它迁移到有限域 $\mathbb{F}_q$ 上来，于是问题的关键便在于，在什么情况下存在环同构 $\mathbb{F}_q[X]/\langle X^n-1\rangle \xrightarrow{\sim} \mathbb{F}_q^n$。

设 $\omega_n$ 表 $n$ 次单位根，我们有 $\mathbb{C}[X]/\langle X^n-1\rangle \xrightarrow{\sim} \prod_1^n \mathbb{C}[X]/\langle X-\omega_n^j\rangle$，这就给出了 DFT 所依据的环同构。循这种思路，我们给出如下定理。

**定理 3.3.1 (中国剩余定理)** 设 $R$ 为环，$I_1,\cdots,I_n$ 为一族理想。假设对每个 $i\neq j$ 皆有 $I_i+I_j=R$，则环同态

$$\begin{aligned}\varphi: R &\longrightarrow \prod_{i=1}^n R/I_i,\\ r &\longmapsto (r \bmod I_i)_1^n\end{aligned}$$

诱导出环同构

$$R\Big/\left(\bigcap_{i=1}^n I_i\right) \xrightarrow{\sim} \prod_{i=1}^n R/I_i.$$

于是，问题的关键在于多项式 $X^n-1$ 能否在 $\mathbb{F}_q$ 中分解为 $n$ 个不同一次因式的乘积，而这又导致我们在 $\mathbb{F}_q$ 中寻找 $n$ 次本原单位根。若存在，则由中国剩余定理，我们可用这个单位根在该有限域上做傅里叶变换。

**定义 3.3.2** 设 $G$ 是群，$F$ 是域。

- 设 $x\in G$。定义 $x$ 的阶 $\operatorname{ord}(x) := |\langle x\rangle|$，即其生成群的元素个数。
- 若 $g\in F^\times$ 满足 $\langle g\rangle = F^\times$，即 $g$ 是 $F$ 的乘法群的生成元，则称 $g$ 为 $F$ 的**原根**。

特别地，我们考虑 $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$，其中 $p$ 为素数，其乘法群的阶为 $p-1$。对于 $k\leq p$，我们有 $k^{\varphi(p)} = k^{p-1} = 1$。因此在数论中我们将满足 $(g) = \varphi(m)$ 的 $g$ 称作模 $m$ 的原根。

设 $g$ 为 $\mathbb{F}_p$ 的一个原根，则 $g$ 是 $\mathbb{F}_p$ 的一个 $p-1$ 次本原单位根，因之对任意 $d\mid p-1$ 都有 $g^d$ 是 $\mathbb{F}_p$ 的 $(p-1)/d$ 次本原单位根。由此，我们将 $\mathbb{F}_p$ 上的傅里叶变换转化为了求原根的问题。

当 $g$ 为 $\mathbb{F}_p$ 的一个原根时，为了将傅里叶变换迁移到 $\mathbb{F}_p$ 上，我们首先要求 $p-1$ 含有因子 $2^m$。只有这样，我们才能对所有 $n=2^k\,(k\leq m)$ 做长度为 $n$ 的傅里叶变换，其具体做法是用 $g^{(p-1)/n}$ 替代 $\omega_n$。

我们还希望 $m$ 尽可能大。基于这点，最常用的 $p$ 是 998244353，它等于 $119\cdot 2^{23}+1$。

```
#define ll long long
const ll mod = 998244353, G = 3;
const ll invG = Qpow(G, mod - 2);
const ll inv2 = Qpow(2, mod - 2), invI = Qpow(I, mod - 2);
#define ck(x) ((x)>=mod?(x)-mod:(x))

void NTT(ll *f, ll flag, ll n) {
    for (ll i = 0;i<n;i++)
    if (i < tr[i]) swap(f[i], f[tr[i]]);
    for (ll p = 2; p <= n; p <<= 1) {
        ll len = p >> 1, wn = Qpow(flag ? G : invG, (mod - 1) / p
           );
        for (ll k = 0, buf = 1; k < n; k += p, buf = 1)
            for (ll i = k, tmp; i < k + len; i++)
                tmp = buf * f[i + len] % mod,
                f[i + len] = ck(f[i] - tmp + mod),
                f[i] = ck(f[i] + tmp), buf = buf * wn % mod;
    }
    if (!flag) {
        ll invn = Qpow(n, mod - 2);
        for (ll i = 0; i < n; i++) f[i] = f[i] * invn % mod;
    }
}

while (n < m) n <<= 1;
for (ll i = 0; i < n; i++)
    tr[i] = (tr[i >> 1] >> 1) | ((i & 1) ? n >> 1 : 0);
```

上面这个 $\mathbb{F}_p$ 上的快速傅里叶变换算法又称快速数论变换 (NTT)。

## 3.4 有限域乘法的快速算法

现在考虑一般的有限域 $\mathbb{F}_q$，其中 $q=p^m$，$p$ 为素数。留意到 $\mathbb{F}_q$ 中的元素无非是 $\mathbb{F}_p$ 上次数小于 $m$ 的多项式加上一个 $m$ 次首一不可约多项式为零的限制。

在本原单位根的次数足够大时，傅里叶变换可以处理多项式乘法。因此上述 NTT 算法给出了一个在 $O(n\log n)$ 的时间内计算 $\mathbb{F}_q[X]$ 中次数为 $n$ 的两个多项式乘法的方法。

```
void Mul(ll *f, ll *g, ll *p, ll n, ll m){
    static ll _f[N], _g[N], _n, _m;
    _n = n, _m = m, m += n, n = 1;
    while (n < m) n <<= 1;
```

```
    for (ll i = 0; i < n; i++)
        tr[i] = (tr[i >> 1] >> 1) | ((i & 1) ? n >> 1 : 0);
    for (ll i = 0; i < _n; i++) _f[i] = f[i];
    for (ll i = 0; i < _m; i++) _g[i] = g[i];
    for (ll i = _n; i < n; i++) _f[i] = 0;
    for (ll i = _m; i < n; i++) _g[i]=0;
    NTT(_f, 1, n), NTT(_g, 1, n);
    for (ll i = 0; i < n; i++) p[i] = _f[i] * _g[i] % mod;
    NTT(p, 0, n);
}
```

然而，现在我们还不能快速计算有限域中的乘法，因为得出多项式后，我们还需对一个首一不可约多项式取模。我们将这个问题留到下一节处理。

# 4 形式幂级数环与 Lagrange 反演

## 4.1 收敛性与级数复合

设环 $R$ 自带拓扑，兹定义形式幂级数列收敛的概念。记 $[X^n]f(X)$ 表示 $X^n$ 项系数 $c_n$。

注意，下面定义的收敛模式只适用于本文。一般地，在拓扑的环（如复数域）的形式幂级数环上给出一个良好的拓扑，这是一个尚未解决的问题。

**定义 4.1.1**

- 一元形式幂级数列 $\{f_n(X)\}$ 若满足各次项系数的序列 $\{[X^k]f_n(X)\}$ 各自收敛于 $[X^k]f(X)$，则称 $\{f_n(X)\}$ 收敛于 $f(X)$，记作 $\lim f_n(X) = f(X)$。
- 若 $\lim \sum_{i\le n} f_i(X) = f(X)$，则称该无穷和等于 $f(X)$。相应可定义无穷乘积 $\prod_i f_i(X)$。
- 设一元形式幂级数 $f(X), g(X)$ 的系数为 $\{f_n\}$ 与 $\{g_n\}$。其复合 $(f\circ g)(X) := f(g(X))$ 若良定义，则是收敛的无穷和 $\sum_0^\infty f_n g(X)^n$。

在一般的拓扑环 $R$ 上，形式幂级数收敛相当于是系数逐点收敛。当 $R$ 是有限域 $\mathbb{F}_q$ 时则更为简单，因为此时系数的拓扑只能取离散拓扑，收敛式 $f_n \to f$ 即表示 $\operatorname{ord}(f_n - f) \to \infty$。另一方面，若复合 $f\circ g$ 良定义，则它们满足以下几个条件中至少一条：

- $f$ 仅有有限项非零，即 $f$ 是多项式。
- $[X^0]g(X) = 0$。
- $f$ 和 $g$ 是能使对应无穷和的系数逐项在拓扑环 $R$ 上收敛的特殊级数。

零次项为零的形式幂级数 $f, g \in XR[\![X]\!]$ 的复合 $f\circ g$ 及 $g\circ f$ 一定收敛在 $XR[\![X]\!]$ 中。确切地说，$XR[\![X]\!]$ 对复合 $\circ$ 成一以 $X$ 为单位元的幺半群 $(XR[\![X]\!], \circ)$。

为了求解复合逆，我们需要引入 Lagrange 反演。接下来我们用 $\mathbb{F}$ 表示域。

## 4.2 Lagrange反演

### 4.2.1 一般形式及其推广

**引理 4.2.1.1 (形式留数)** 设 $F(X)=\sum_{n\geq 1}a_nX^n$ 是 $\mathbb{F}$ 上的 Laurent 级数，$[X^1]F(X)\neq 0$，则有

$$\left[X^{-1}\right]\left(F'(X)F(X)^k\right)=[k=-1].$$

**定理 4.2.1.2** 设 $F(X)\in\mathbb{F}(\!(X)\!)$，则有

$$n[X^n]F^{\langle -1\rangle}(X)^k=k\left[X^{n-k}\right]\left(\frac{X}{F(X)}\right)^n=k\left[X^{-k}\right]F(X)^{-n}.$$

**推论 4.2.1.3** 设 $f(X),G(X)\in\mathbb{F}(\!(X)\!)$。若 $f(X)=XG(f(X))$，则

$$n[X^n]f(X)^k=k\left[X^{n-k}\right]G(X)^n.$$

**证明 (形式幂级数情形)** 令 $F^{\langle -1\rangle}(X)^k=\sum_{i\geqslant k}p_iX^i$，则

$$X^k=F^{\langle -1\rangle}(F(X))^k=\sum_{i\geqslant k}p_iF(X)^i.$$

等式两端取微分得

$$\begin{aligned}
kX^{k-1}&=\sum_{i\geqslant k}ip_iF(X)^{i-1}F'(X),\\
\frac{kX^{k-1}}{F(X)^n}&=\sum_{i\geqslant k}ip_iF(X)^{i-n-1}F'(X)\\
&=np_nF(X)^{-1}F'(X)+\sum_{i\geqslant k,i\neq n}ip_i\frac{1}{i-n}\frac{d}{dX}F(X)^{i-n},
\end{aligned}$$

于是

$$\begin{aligned}
\left[X^{-1}\right]\frac{kX^{k-1}}{F(X)^n}&=\left[X^{-1}\right]np_nF(X)^{-1}F'(X)\\
&=\left[X^{-1}\right]np_n\cdot\frac{a_1+2a_2X+\cdots}{a_1X+a_2X^2+\cdots}\\
&=\left[X^{-1}\right]np_n\left(\frac{1}{X}+\cdots\right)\\
&=np_n.
\end{aligned}$$

从而立即得到

$$\left[X^{-1}\right]\frac{kX^{k-1}}{F(X)^n}=np_n=n[X^n]F^{\langle -1\rangle}(X)^k.$$

□

**命题 4.2.1.4** 设 $F(X)=\sum_{n\geq 1}a_nX^n,H(X)\in\mathbb{F}(\!(X)\!)$，并设 $G(X)$ 是 $F(X)$ 的复合逆，则

$$[X^n]H(F(X))=\frac{1}{n}[X^{n-1}]H'(X)\left(\frac{X}{G(X)}\right)^n.$$

**证明** $H(X)=X^k$ 时就是普通的Lagrange反演；其余情形的 $H(X)$ 乃是 $X^k$ 的线性组合，故只需确认 $H(X)=X^k$ 即完成证明。 □

**命题 4.2.1.4′** 设 $F(X)=\sum_{n\geq 1}a_nX^n, G(X), H(X)\in\mathbb{F}(\!(X)\!)$ 满足 $F(G(X))=H(X)$，那么

$$[X^n]F(X)=\frac{1}{n}[X^{n-1}]H'(X)\left(\frac{X}{G(X)}\right)^n.$$

或者

$$F(X)=\frac{d}{dX}\left[H'(X)\left(\frac{X}{G(X)}\right)^n\right].$$

**证明** 无非令新的 $F(X)$ 代表原先的 $H(F(X))$，则 $F(G(X))=H(X)$。 □

推论 **4.2.1.3** 也有相应的复合形式。

**命题 4.2.1.5** 设 $f(X), G(X), H(X)\in\mathbb{F}(\!(X)\!)$。若 $f(X)=XG(f(X))$，则

$$n[X^n]H(f(X))=\frac{1}{n}\left[X^{n-1}\right]H'(X)G(X)^n.$$

### 4.2.2 另类Lagrange反演

**定理 4.2.2.1** 设 $F(X)=\sum_{n\geq 1}a_nX^n\in\mathbb{F}(\!(X)\!)$，则有

$$[X^n]F^{\langle -1\rangle}(X)^k=\left[X^{n-k}\right]F'(X)\left(\frac{X}{F(X)}\right)^{n+1}=\left[X^{-k-1}\right]F'(X)F(X)^{-n-1}.$$

**证明** 考虑在 (4.3) 式中不进行求导，而是乘以 $F'(X)F(X)^{-n-1}$。运用引理 **4.2.1.4** 得

$$\begin{aligned}
X^k&=\sum_{i\geqslant k}p_iF(X)^i,\\
X^kF'(X)F(X)^{-n-1}&=\sum_{i\geqslant k}p_iF'(X)F(X)^{i-n-1},\\
\left[X^{-1}\right]\left(X^kF'(X)F(X)^{-n-1}\right)&=\left[X^{-1}\right]\sum_{i\geqslant k}p_iF'(X)F(X)^{i-n-1}=p_n,\\
\left[X^{-k-1}\right]kF'(X)F(X)^{-n-1}&=[X^n]F^{\langle -1\rangle}(X)^k.
\end{aligned}$$

□

另类Lagrange反演也有相应的复合形式。

**定理 4.2.2.2** 设 $F(X)=\sum_{n\geq 1}a_nX^n, H(X)\in\mathbb{F}(\!(X)\!)$，并设 $G(X)$ 是 $F(X)$ 的复合逆，则

$$[X^n]H(F(X))=[X^n]H(X)G'(X)\left(\frac{X}{G(X)}\right)^{n+1}=\left[X^{-1}\right]H(X)G'(X)G(X)^{-n-1}.$$

**定理 4.2.2.2′** 设 $F(X)=\sum_{n\geq 1}a_nX^n, G(X), H(X)\in\mathbb{F}(\!(X)\!)$ 满足 $F(G(X))=H(X)$，那么

$$[X^n]F(X)=[X^n]H(X)G'(X)\left(\frac{X}{G(X)}\right)^{n+1}=\left[X^{-1}\right]H(X)G'(X)G(X)^{-n-1}.$$

# 5 $\mathbb{F}_p[\![X]\!]$ 上的快速算法

考虑形式幂级数环 $\mathbb{F}_p[\![X]\!]$，其中 $p$ 为素数。

## 5.1 初等函数

**定理 5.1.1** 以下 $\mathbb{F}_p[\![X]\!]$ 中的运算可在 $O(n\log n)$ 时间内完成：

- 求逆：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $f^{-1}\in\mathbb{F}_p[\![X]\!]$ 使得 $ff^{-1}=1$。
- 带余除法：对于 $f,g\in\mathbb{F}_p[X]$，求 $h,r\in\mathbb{F}_p[X]$ 使得 $f=hg+r$。
- 指数：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $\exp(f)\in\mathbb{F}_p[\![X]\!]$。
- 对数：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $\log(f)\in\mathbb{F}_p[\![X]\!]$。
- 幂：对于 $f\in\mathbb{F}_p[\![X]\!],m\in\mathbb{N}$，求 $f^m\in\mathbb{F}_p[\![X]\!]$。
- 开根：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $\sqrt{f}\in\mathbb{F}_p[\![X]\!]$。
- 三角函数：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $\sin(f),\cos(f),\tan(f)$。
- 反三角函数：对于 $f\in\mathbb{F}_p[\![X]\!]$，求 $\arcsin(f),\arccos(f),\arctan(f)$。

我们有必要对此进行进一步说明。最基本地，指数函数与对数函数定义为

$$\exp(X):=\sum_{n=0}^{\infty}\frac{X^n}{n!},\qquad \log(1-X):=-\sum_{n=1}^{\infty}\frac{X^n}{n},$$

其中 $1/n!$ 以及 $1/n$ 皆在 $\mathbb{F}_p$ 中定义。定理中给出的 exp 与 log 无非是形式幂级数的复合。进一步，令 $i$ 指代 $x^2\equiv -1 \pmod p$ 的解，则我们定义三角幂级数

$$\cos(X):=\frac{\exp(iX)+\exp(-iX)}{2},\quad \sin(X):=\frac{\exp(iX)-\exp(-iX)}{2i},\quad \tan(X)=\frac{\sin(X)}{\cos(X)}.$$

最后，我们来进一步说明多项式在 $\mathbb{F}_p[\![X]\!]$ 中的求逆。设 $f=\sum_0^\infty a_jX^j, g=\sum_0^\infty b_kX^k$，等式 $fg=1$ 给出方程组

$$\begin{aligned}
&1=a_0+b_0,\\
&0=a_1b_0+a_0b_1,\\
&0=a_2b_0+a_1b_1+a_0b_2,\\
&\quad\cdots\\
&0=\sum_{j=0}^{n}a_jb_{n-j},\qquad\cdots\qquad 0=\sum_{j=0}^{n}a_jb_{m-j},\\
&\quad\cdots
\end{aligned}$$

每一个第 $m$ 行都仅有 $b_m$ 一个新的未知数。从 $m=n+1$ 开始，其方程总是

$$b_m = -a_0^{-1}\sum_{j=0}^{m-1} a_j b_{m-j},$$

这是一个常系数齐次线性递推。反过来，所有常系数齐次线性递推都对应一个多项式的求逆。

### 5.2 复合逆的快速算法

对于 $g(f(X))=X$，首先有 Lagrange 反演

$$[X^n]g(X) = \frac{1}{n}\left[X^{n-1}\right]\left(\frac{X}{f(X)}\right)^n,$$

据此就有

$$\begin{aligned}
g(X) &= \sum_{k=0}^{n}\left(\frac{1}{k}\left[X^{k-1}\right]\left(\frac{X}{f(X)}\right)^k\right)X^k \\
&= \sum_{i=0}^{L}\sum_{j=0}^{L}\left(\frac{1}{iL+j}\left[X^{iL+j-1}\right]\left(\frac{X}{f(X)}\right)^{iL+j}\right)X^{iL+j} \\
&= \sum_{i=0}^{L}X^{iL}\sum_{j=0}^{L}\frac{X^j}{iL+j}\left[X^{iL+j-1}\right]\left(\left(\frac{X}{f(X)}\right)^{iL}\left(\frac{X}{f(X)}\right)^{j}\right),
\end{aligned}$$

于是 $O(n\sqrt{n}\log n)$ 预处理 $\left(\frac{X}{f(X)}\right)^{iL}$ 与 $\left(\frac{X}{f(X)}\right)^{j}$ 即可。内层和式暴力计算并与 $X^{iL}$ 相乘，相加即是答案，这部分复杂度为 $O(n^2)$。

### 5.3 多项式复合

使用与复合逆类似的做法。首先观察到

$$\begin{aligned}
\sum_{i=0}^{n}\left([X^i]f(X)\right)g(X)^i &= \sum_{i=0}^{L}\sum_{j=0}^{L}\left(\left[X^{iL+j}\right]f(X)\right)g(X)^{iL+j} \\
&= \sum_{i=0}^{L}g(X)^{iL}\sum_{j=0}^{L}\left(\left[X^{iL+j}\right]f(X)\right)g(X)^{j},
\end{aligned}$$

于是 $O(n\sqrt{n}\log n)$ 预处理 $g(X)^{iL}$ 与 $g(X)^i$ 即可。

$\alpha$ 贾岚晴