CSS.307.1: Algebra, Number Theory and Computation

Instructor: Mrinal Kumar

TIFR 2025, Aug-Dec

Scribe: Soham Chatterjee

soham.chatterjee@tifr.res.in
Website: sohamch08.github.io

Contents

Chapter 1

Polynomial Arithmetic

1.1 Multiplication

1.2 Fast Division

Polynomial Division
Input: $f, g \in \mathbb{F}[X]$, $\deg(f, g) \leq d$
Output: Quotient and reminder when f is divided by g.

Suppose $\deg f = a$ and $\deg g = b$. Let $(q, r) \in \mathbb{F}[X]$ are the quotient and remainder when f is divided by g i.e. $f = qg + r$. Therefore $\deg q = a - b$ and $m := \deg r < b$.

We can follow the long division algorithm to find (q, r). This algorithm takes $O(a - b) = O(d)$ many iteration to find q. And in each iteration we subtract a polynomial from another polynomial by multiplying one of them with power of x. For the multiplying with power x is just shifting of the coefficients. For the subtraction of polynomials it takes $O(d)$ time. Therefore each iteration of the algorithm takes $O(d)$ time complexity. Therefore the long division algorithm takes $O(d^2)$ time complexity.

If we can obtain q from f, g then we can get r by following the equation $r = f - gq$.

1.2.1 Reversal of Polynomials

Idea. *Reversal of Polynomials i.e. if $f \in \mathbb{F}[X]$ such that $f = f_0 + f_1X + \cdots + f_aX^a$ then*

$$rev(f) = f_0X^a + f_1X^{a-1} + \cdots + f_a = f\left(\frac{1}{X}\right)X^a$$

Note:-

We have $\deg f \geq \deg(rev(f))$. Degree of $rev(f)$ can be strictly lesser than the degree of f. For example if $f_0 = 0$ and $f_1 \neq 0$, since $rev(f) = X^a f\left(\frac{1}{X}\right)$ the degree of $rev(f)$ is $a - 1$.

So using reversal we will review the equation $f = gq + r$:

$$\begin{aligned}
& f = qg + r \\
\iff & X^a f\left(\frac{1}{X}\right) = X^a\left[q\left(\frac{1}{X}\right)g\left(\frac{1}{X}\right) + r\left(\frac{1}{X}\right)\right] \\
\iff & X^a f\left(\frac{1}{X}\right) = cdX^a q\left(\frac{1}{X}\right)g\left(\frac{1}{X}\right) + X^a r\left(\frac{1}{X}\right) \\
\iff & rev(f) = rev(q)rev(g) + X^{a-m}rev(r)
\end{aligned}$$

Now we know $a \geq b > m \implies a - m \geq b - m > 0$. Therefore $X^{a-m}rev(r)$ is multiple of some nontrivial power of X. Now also we have

$$a - m > a - b = \deg q \geq \deg(rev(q))$$

Therefore we have

$$rev(f) \equiv rev(q)rev(g) \bmod X^{a-m}$$

Since $a - m \geq a - b + 1$ we have

$$rev(q) \bmod X^{a-m} \equiv rev(q) \bmod X^{a-b+1} \equiv rev(q)$$

Therefore we have

$$rev(f) \equiv rev(q)rev(g) \bmod X^{a-b+1}$$

Hence it suffices to recover $rev(q)$ in order to recover q from here. So the problem now reduced to finding a solution $h \in \mathbb{F}[X]$ for the system $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$.

1.2.2 Find solution of $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

Solve $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$
Input: $\tilde{f}, \tilde{g} \in \mathbb{F}[X]$, $\deg(f,g) \leq d$, $\tilde{f}(0), \tilde{g}(0) \neq 0$ with $N \in \mathbb{N}$
Output: Find solution h for the equation $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

Lemma 1.2.1
There is an unique $h \in \mathbb{F}[X]$ satisfying $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$.

Proof: Let $\deg \tilde{f} = k$ and $\deg \tilde{g} = l$. Then Suppose $\tilde{f} = \sum_{i=0}^{k} \tilde{f}_i X^i$ and $\tilde{g} = \sum_{i=0}^{l} \tilde{g}_i X^i$. Then we can write the equation $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$ as a linear system like the following:

$$\begin{bmatrix} \tilde{g}_0 & & & & \\ \tilde{g}_1 & \tilde{g}_0 & & & \\ \tilde{g}_2 & \tilde{g}_1 & \tilde{g}_0 & & \\ \vdots & & & \ddots & \\ & & & & \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{k-l} \end{bmatrix} = \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \vdots \\ \tilde{f}_k \end{bmatrix}$$

Lets call the matrix G. Since $\tilde{g}_0 \neq 0$ the G has nonzero elements in the diagonal. Since the G is lower triangular the determinant of the G is nonzero. Therefore there exists unique solution solution for h. ■

But we don't know how to find inverse of G in near linear time. So we cannot find h like this.

Idea. *Find a power series solution for* $h = \frac{\tilde{f}}{\tilde{g}} \bmod X^N$ *in* $\mathbb{F}[\![X]\!] \supseteq \mathbb{F}[X]$ *since in* $\mathbb{F}[\![X]\!]$ *inverse of* $\tilde{g}$ *exists*

Lemma 1.2.2
For every power series $P = \sum_{i=0}^{\infty} P_i X^i \in \mathbb{F}[\![X]\!]$, P has a multiplicative inverse iff $P_0 \neq 0$.

Since we are dealing with the equation $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$ and $\tilde{g}(0) \neq 0$ there exists a power series solution for h. We will see two algorithms to find $h \in \mathbb{F}[\![X]\!]$.

1.2.2.1 Algorithm I

$$\frac{\tilde{f}(X)}{\tilde{g}(X)} \bmod X^N = \frac{\tilde{f}(X)}{\tilde{g}(X)} \frac{\tilde{g}(-X)}{\tilde{g}(-X)} \bmod X^N = \frac{\tilde{f}(X)\tilde{g}(-X)}{\tilde{g}(X)\tilde{g}(-X)} \bmod X^N$$

Now $\tilde{g}(X)\tilde{g}(-X)$ is an even function. Therefore $\exists\, G \in \mathbb{F}[X]$ and $\deg G \le d$ such that $G(X^2) = \tilde{g}(X)\tilde{g}(-X)$. Now we can also decompose $\tilde{f}(X) = \tilde{f}_0(X^2) + X\tilde{f}_1(X^2)$ and $\tilde{g}(-X) = \tilde{g}_0(X^2) + X\tilde{g}_0(X^2)$. Then we have

$$\begin{aligned}
\tilde{f}(X)\tilde{g}(-X) &= \left(\tilde{f}_0(X^2) + X\tilde{f}_1(X^2)\right)\left(\tilde{g}_0(X^2) + X\tilde{g}_1(X^2)\right)\\
&= \underbrace{\left[\tilde{f}_0(X^2)\tilde{g}_0(X^2) + X^2\tilde{f}_1(X^2)\tilde{g}_1(X^2)\right]}_{F_0(X^2)} + X\underbrace{\left[\tilde{f}_1(X^2)\tilde{g}_0(X^2) + \tilde{f}_0(X^2)\tilde{g}_1(X^2)\right]}_{F_1(X^2)}\\
&= F_0(X^2) + XF_1(X^2) && [\deg F_i \le d\ \forall i \in \{0,1\}]
\end{aligned}$$

Therefore we have

$$\begin{aligned}
\frac{\tilde{f}(X)}{\tilde{g}(X)} \bmod X^N &= \frac{F_0(X^2)}{G(X^2)} + X\frac{F_1(X^2)}{G(X^2)} \bmod X^N\\
&= \underbrace{\frac{F_0(X^2)}{G(X^2)} \bmod X^N}_{\frac{F_0(Z)}{G(Z)} \bmod Z^{\frac{N}{2}}\big|_{Z=X^2}} + X\underbrace{\frac{F_0(X^2)}{G(X^2)} \bmod X^N}_{\frac{F_1(Z)}{G(Z)} \bmod Z^{\frac{N}{2}}\big|_{Z=X^2}}
\end{aligned}$$

Now we recurse. So the algorithm is

Algorithm 1: Solve $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

Input: $\tilde{f}, \tilde{g} \in \mathbb{F}[X]$, $\deg(f,g) \le d$, $\tilde{f}(0), \tilde{g}(0) \ne 0$ with $N \in \mathbb{N}$
Output: Find solution h for the equation $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

begin
  Construct G, F_0, F_1
  Compute $\frac{F_0(Z)}{G(Z)} \bmod Z^{\frac{N}{2}}$, $\frac{F_1(Z)}{G(Z)} \bmod Z^{\frac{N}{2}}$
  Set $Z \longleftarrow X^2$ and combine and return

Now we have that

$$\frac{F_1(Z)}{G(Z)} \bmod Z^{\frac{N}{2}} = \frac{F_1(Z) \bmod Z^{\frac{N}{2}}}{G(Z) \bmod Z^{\frac{N}{2}}} \bmod X^{\frac{N}{2}}$$

Hence the degree got reduced by half. So in the recursion step we can reduce the degree with this.

Time Complexity: If $T(N)$ is the total time taken while solving for modulo X^N then we have the recursion relation

$$T(N) \le 2T\left(\frac{N}{2}\right) + 10M(N)$$

Hence the total running time of this algorithm is $T(N) = M(N)\log N = N poly(\log N)$

1.2.2.2 Algorithm II

Here we can divide h into two parts with each part of degrees $< \frac{N}{2}$. Then $h(X) = h_0(X) + X^{\frac{N}{2}}h_1(X)$ where $\deg h_0 < \frac{N}{2}$ and $\deg h_1 < \frac{N}{2}$. Then we have

$$\tilde{f} - \left(h_0 + h_1X^{\frac{N}{2}}\right)\tilde{g} \equiv 0 \bmod X^N \implies (\tilde{f} - h_0\tilde{g}) - X^{\frac{N}{2}}h_1\tilde{g} \equiv 0 \bmod X^N$$

Hence we have $\tilde{f} - h_0\tilde{g} \equiv 0 \bmod X^{\frac{N}{2}}$. Therefore we have

$$X^{\frac{N}{2}} \mid \tilde{f} - h_0\tilde{g} \implies \tilde{f} - h_0\tilde{g} = X^{\frac{N}{2}}$$

Hence

$$X^{\frac{N}{2}}p - X^{\frac{N}{2}}h_1\tilde{g} \equiv 0 \bmod X^N \implies p - h_1\tilde{g} \equiv 0 \bmod X^{\frac{N}{2}}$$

Therefore we have the following algorithm

Algorithm 2: Solve $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

Input: $\tilde{f}, \tilde{g} \in \mathbb{F}[X]$, $\deg(f,g) \leq d$, $\tilde{f}(0), \tilde{g}(0) \neq 0$ with $N \in \mathbb{N}$
Output: Find solution h for the equation $\tilde{f} - h\tilde{g} \equiv 0 \bmod X^N$

```
begin
    Construct h_0, h_1
    Solve f~ - h_0 g~ ≡ 0 mod X^(N/2)
    R ⟵ (f~ - h_0 g~) / X^(N/2)
    Solve R - h_1 g~ ≡ 0 mod X^(N/2)
    Output h_0 + X^(N/2) h_1
```

Time Complexity: If $T(N)$ is the total time taken while solving for modulo X^N then we have the recursion relation

$$T(N) \leq 2T\left(\frac{N}{2}\right) + O(M(N))$$

Hence the total running time of this algorithm is $T(N) = O(M(N)\log N) = N poly(\log N)$

1.3 Chinese Remainder Theorem

1.4 Derivatives

Chapter 2

Greatest Common Divisor

2.1 Fast Parallel GCD

2.2 Resultants

Chapter 3

Modular Composition

Chapter 4

Univariate Polynomial Factorization

4.1 Cantor-Zassenhaus

4.2 Barlekamp

Chapter 5

Bivariate Polynomial Factorization