Fast Division in F[x]

October 17, 2023 9:16 AM

Assignment #3 is due on Monday @ 11pm.

Let $a, b \in F[x]$, $a = a_m x^m + \cdots + a_1 x + a_0$ and $b = b_n x^n + \cdots + b_1 x + b_0$ and $m \geq n$.

Let $\boxed{a = bq + r}$ with $r = 0$ or $\deg r < \deg b$. $a \div b$.

The classical $\div$ algorithm does $(m-n+1)\cdot n$ mults in F.

If $m = 2n \Rightarrow (n+1)\cdot n \in O(n^2)$.

1. Compute q. ← use the FFT.
2. Compute $r = a - b \cdot q$ with a fast $\times$.

Define $a^r = a_0 x^m + a_1 x^{m-1} + \cdots + a_m$ the reciprocal polynomial.

Idea 1 Compute $q^r = \frac{a^r}{b^r}$ truncated to $O(x^{\boxed{m-n}+1})$, $m-n = \deg(q)$.

Then $q = (q^r)^r$.

Example $a = 6x^2 + 8x + 2$ $\qquad a^r = 6 + 8x + 2x^2$ $\qquad \deg q = 1$

$b = 2x + 4$ $\qquad b^r = 2 + 4x$

$$
\begin{array}{r l}
 & 3 - 2x + 5x^2 \\
2 + 4x \;) & 6 + 8x + 2x^2 \\
 & -(6 + 12x) \\
 & \quad -4x + 2x^2 \\
 & \quad -(-4x - 8x^2) \\
 & \quad\quad 0 + 10x^2
\end{array}
$$

$q^r = 3 - 2x \Rightarrow q = 3x - 2.$

This would be $O(n^2)$.

Idea 2 We want to compute $\frac{a^r}{b^r}$.

Compute $\frac{1}{b^r}$ to $O(x^{m-n+1})$ as a power series then.

Compute $q_r = \frac{1}{b^r} \cdot a^r$ to $O(x^{m-n+1})$ using a second fast $\times$ in $F[x]$.

$$
\begin{array}{r l}
 & \frac{1}{2} - x + 2x^2 \\
b^r = 2 + 4x \;) & 1 \\
 & -(1 + 2x) \\
 & \quad -2x \\
 & \quad -(-2x - 4x^2) \\
 & \quad\quad 0 + 4x^2
\end{array}
$$

$q^r = \frac{1}{b^r} \cdot a^r = (\frac{1}{2} - x)(\underbrace{6 + 8x + \cdots}_{a^r})$

$= 3 + 4x - 6x + \cdots$

$= 3 - 2x.$

This would also cost $O(n^2)$.

$0 + 4x^{-}$ [illegible]

This would also cost $O(n^2)$. ☹

Theorem 9.2 (Modern Computer Algebra).

Let R be a comm. ring with 1_R. ($R = F$ for us).
Let $f \in R[x]$. $(f = f_0 + f_1 x + \cdots)$ with $f_0^{-1} \in R$.

Let $\underline{y_0 = f_0^{-1}}$ and $y_i = 2y_{i-1} - f \cdot y_{i-1}^2 \bmod x^{2^i}$ for $i > 0$.

Then $f \cdot y_i \equiv 1 \bmod x^{2^i}$ for $i \geq 0$.

$y_i = f^{-1}$ up to $O(x^{2^i})$.

Proof. By induction on i. We will prove $1 - f y_i \equiv 0 \bmod x^{2^i}$.

✓ $i = 0$: $1 - f \cdot y_0 = 1 - (f_0 + f_1 x + \cdots) \cdot \frac{1}{f_0} = 0 + \cdot x + \cdots \equiv 0 \bmod x^1$

$i > 0$:
$$1 - f y_i = 1 - f(2y_{i-1} - f y_{i-1}^2)$$
$$= 1 - 2f y_{i-1} + f^2 y_{i-1}^2$$
$$= (1 - f y_{i-1})^2$$

By induction on i: $1 - f \cdot y_{i-1} \equiv 0 \bmod x^{2^{i-1}}$
$$(1 - f y_{i-1})^2 = (0 + 0x + \cdots + 0 \cdot x^{2^{i-1}-1} + \cdot x^{2^{i-1}} + \cdots +)^2$$
$$= \cdot x^{2^i} + \cdot x^{2^i+1} + \cdots$$
$$= 0 \bmod x^{2^i}.$$

Recall the Newton iteration to solve $f(y) = 0$.

y_0 = initial approx.
$y_{k+1} = y_k - f(y_k)/f'(y_k)$.

$f(y) = 0 \Rightarrow b = \frac{1}{y} \Rightarrow y = \frac{1}{b}$.

To compute $y = \frac{1}{b}$ use $f(y) = b - \frac{1}{y} = b - y^{-1}$

$f'(y) = \frac{1}{y^2}$

$$y_{k+1} = y_k - \frac{b - \frac{1}{y_k}}{\frac{1}{y_k^2}} = y_k - b y_k^2 + y_k = 2y_k - b y_k^2.$$

Expand (y_k^2) mod p.

[illegible] $b y_k^2$ [illegible] no divisions.

y_k Expand y_k ...

$$\Longrightarrow \quad y_{k+1} = 2y_k - b\, y_k^2$$ There are no divisions.

two multiplications.

$$y_0 = \frac{1}{b_0}.$$

Example. Compute $\frac{1}{1-x+x^2 = b^r} \bmod x^4$.

$i=0 \quad y_0 = \frac{1}{b^r} \bmod x = \frac{1}{1} \bmod x^{2^0} = 1.$

$i=1 \quad y_1 = 2y_0 - b^r \cdot y_0^2 \bmod x^2$

$= 2\cdot 1 - (1-x+x^2)\cdot 1^2 \bmod x^2$

$= 2 - (1-x)\cdot 1 \bmod x^2$

$= 1 + x.$

$i=2 \quad y_2 = 2y_1 - b^r y_1^2 \bmod x^4$

$= 2\cdot(1+x) - (1-x+x^2)(1+x)^2 \bmod x^4$

$= 2+2x - (1-x+x^2)\cdot(1+2x+x^2) \bmod x^4$

$= 1+x-x^3-x^4 \bmod x^4$

$= 1+x-x^3.$ use a polynomial x

Let $M(n)$ be the # arith ops for multiplying two polynomials of degree $\le n$. [If we use the FFT then $M(n) \in O(n \log n)$].

Let $I(n)$ be the # arith ops for computing $\frac{1}{b^r} \bmod x^n$ using the Newton iteration.

$I(1) = 1 \leftarrow f_0^{-1} = y_0$

$I(n) \le I(\frac{n}{2}) + M(\frac{n}{2}) + M(n) + cn$ ← $2 \cdot y_{i-1}$ and $-$.

recursive call to compute y_{i-1} to $O(x^{n/2})$; for y_{i-1}^2; $b^r \cdot y_{i-1}^2$ to $O(x^n)$.

Exercise. Assume $M(n) > 2M(n/2)$ [Mult in $F[x]$ is super-linear]

Then show that $I(n) < 3M(n) + c'n.$

Let $D(n)$ be the cost of dividing $a(x) \div b(x)$ to get the quotient q and remainder r.

$$D(n) = \underset{\substack{\uparrow \\ (b^r)^{-1}}}{I(n)} + \underset{q^r = \frac{1}{b^r} \cdot a^r}{\underline{M(n)}} + \underset{\substack{\uparrow \\ a - b \cdot q = r}}{\underline{M(n)}} + c \cdot n$$

Then $D(n) = 5M(n) + c \cdot n. \in O(M(n))$.

It is possible using the "middle product" of Zimmermann et.al. to reduce $I(n) < 2\,M(n)$.

Then $D(n) < 4\,M(n)$.