Integer multiplication (contd.)

For today fix $\beta = 2$.

Note that we assumed that $\frac{1}{n}$ and n^{th} roots of unity are available. Today we will show that we can find them while working with integers mod n (for chosen n).

Lemma: Let $n = 2^k$, $a \in \mathbb{Z}$. Then for any $0 < p < n$,

$$\sum_{i=0}^{n-1} a^{ip} = 0 \mod (1 + a^{n/2}).$$

$\leq \frac{n}{2} \log n$

$\hookrightarrow m$.

Facts:

1. $\sum_{i=0}^{n-1} a^i = \prod_{j=0}^{k-1} (a^{2^j} + 1)$

2. If s is odd, $\forall x \in \mathbb{Z}$ $(1+x) \mid (1+x^s)$.

Proof: $\sum_{i=0}^{n-1} a^{ip} = \prod_{j=0}^{k-1} (a^{p \cdot 2^j} + 1)$.

Let $p = 2^t \cdot s$ for some $0 \leq t \leq k-1$. (s: odd)

$$\Rightarrow \sum_{i=0}^{n-1} a^{ip} = \prod_{j=0}^{k-1} (a^{s \cdot 2^t \cdot 2^j} + 1)$$

$(1+x^s) = Q \cdot (1+x) + R$

$x \to -1$: $0 = Q \cdot 0 + R$

Look at the factor for $j = k-t-1$

$$(a^{s \cdot 2^t \cdot 2^{k-t-1}} + 1) = (a^{s \cdot 2^{k-1}} + 1).$$

From fact 2, we get that $(a^{2^{k-1}} + 1) \mid (a^{s \cdot 2^{k-1}} + 1)$.

$$\Rightarrow (a^{2^{k-1}} + 1) \mid \sum_{i=0}^{n-1} a^{ip}.$$

$n = 2^k \Rightarrow \frac{n}{2} = 2^{k-1}$.

$m = (a^{n/2} + 1) = (a^{2^{k-1}} + 1)$

In other words, $\sum_{i=0}^{n-1} a^{ip} \equiv 0 \mod (a^{n/2} + 1)$.

Theorem: Let n and w be powers of 2, $m = 1 + w^{n/2}$. Let R_m be the ring of integers modulo m. Then n^{-1} exists in R_m and w is the n^{th} primitive root of unity.

Proof: Let $w^{n/2} = \alpha n$ (as w and n are powers of 2)

(α: power of 2)

$$(w^{n/2} + 1) - \alpha n = 1 \Rightarrow m - \alpha n = 1.$$

($w^{n/2} + 1 = m$)

$$\Rightarrow (-\alpha) n = 1 \mod m.$$

$-1 \to m-1$, $-2 \to m-2$

$-\alpha$ is n^{-1} in R_m.

Since $m = w^{\frac{n}{2}} + 1$, $w^{\frac{n}{2}} = -1 \pmod m$

$$\Rightarrow w^n = 1 \pmod m.$$

$w = e^{\frac{2\pi i}{n} i}$, $e^{\frac{2\pi i}{n} k}$

Now we need to show that w is primitive: $w^p \neq 1 \pmod m$ $\forall\ 1 \le p \le n-1$. Suppose not.

$w^p = 1 \pmod m$. ↯

$n << m$.

Consider $\sum_{i=0}^{n-1} w^{ip} = \sum_{i=0}^{n-1} (w^p)^i = n \pmod m$. But from previous lemma $\sum_{i=0}^{n-1} w^{ip} = 0 \pmod m$. Contradiction.

Computing integers mod m fastly.

Let $w = 2^r$. $\Rightarrow 2^{rn/2} = -1 \pmod m$.

Given a "large number" say

$rn/2$

$\ldots\ldots x_2 x_1 x_0$.

$rn/2$

Break into blocks of size $\frac{rn}{2}$ bits each.

Let those blocks be $\cdots x_2 x_1 x_0$.

$$X = \sum_{i \geq 0} x_i \cdot 2^{i\left(\frac{rn}{2}\right)} = \sum_{i \geq 0} x_i (-1)^i \pmod{m}.$$

Complexity of computing DFT over R_m.

Recall that

$A^m : R_m^n \to R_m^n$

$$\left[F_n(\vec{a})\right]_i = \begin{cases} \left[F_{\frac{n}{2}}(\vec{C})\right]_{\frac{i}{2}} & \text{if } i \text{ is even} \\ \left[F_{\frac{n}{2}}(\vec{D})\right]_{\frac{i-1}{2}} & \text{if is odd.} \end{cases}$$

$C_k = a_k + a_{k+\frac{n}{2}}$

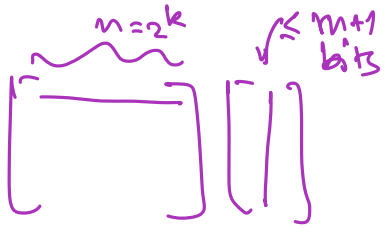
$D_k = \omega^k (a_k - a_{k+\frac{n}{2}})$.

$A^m = \left[\omega^{ij}\right] \begin{bmatrix} a_0 \\ a_{m-1} \end{bmatrix} \in R_m^n$

$a_k + a_{k+\frac{n}{2}}$ ← $O(\frac{rn}{2})$ bit operations.

$\omega^k (a_k - a_{k+\frac{n}{2}})$ ← $O(\frac{rn}{2})$ bit operation for substraction

↑ $O(\frac{rn}{2})$ bit operations for shift.



$\left[F_n(\vec{a})\right]_i$ could have at most $1 + rn + k$ bits. $\leq \frac{3rn}{2}$

{ $n \log n$ operations $\times O(\frac{rn}{2})$

⇒ Each arithmetic operation will cost $O(\frac{rn}{2})$ bit operations

⇒ Bit complexity of DFT over R_m: $O(n^2 \log n \log w)$

⇒ Convolution bit complexity = $O(n \cdot M(n \log n) + n^2 \log n \log w)$.

Obs: If p and q are polynomials with coefs with absolute value at most $\frac{w^{n/4}}{\sqrt{n}}$ can be done in R_m with-out wrap arounds.

Schönhage-Strassen integer multiplication

$$A = A_{b-1} \cdots A_1 A_0$$
$$B = B_{b-1} \cdots B_1 B_0$$

$n = 2^k$

$|A_i| = |B_i| = \ell$ bits.

$\ell = 2^{\lceil \frac{k}{2} \rceil}$ and $b = 2^{\lfloor \frac{k}{2} \rfloor}$.

$\Rightarrow n = b \cdot \ell$.

Note that $A = \sum_{i=1}^{b} A_i \cdot 2^{i\ell}$ and $B = \sum_{i=1}^{b} B_i \cdot 2^{i\ell}$.

Now look at $A \cdot B$. Say $A \cdot B = \sum_{j=0}^{2b-1} y_j \cdot 2^{j\ell}$.

$$y_j = \begin{cases} \sum_{i=0}^{j} A_i B_{j-i} & \text{if } j \le b-1, \\ \sum_{i=j-(b-1)}^{b-1} A_i B_{j-i} & \text{otherwise.} \end{cases}$$

$2^n = -1 \pmod{(2^n+1)}$ and $2^{b\ell} = 2^n$.

If we want to compute $A \cdot B \pmod{(2^n+1)}$;

$$A \cdot B = \sum_{j=0}^{b-1} (y_j - y_{b+j}) \cdot 2^{j\ell} \pmod{(2^n+1)}.$$

Note that $(y_0, \ldots, y_{2b-1}) = (A_0, \ldots, A_{b-1}) * (B_0, \ldots, B_{b-1})$.

Let $z_j = (y_j - y_{b+j}) \ \forall j$.

$$(z_0, \dots, z_{b-1}) = (A_0, \dots, A_{b-1}) *_- (B_0, \dots, B_{b-1})$$

↖ Negatively wrapped convolution.

$$-(b-j-1)\cdot 2^{2l} \le z_j \le (j+1)\,2^{2l} \Rightarrow |\text{Range}| \le b \cdot 2^{2l}.$$

$\Rightarrow$ Computing $z_j \bmod b\,2^{2l}$ will not lose any inf.

↓ Rather we can do

$$z_j \bmod b(2^{2l}+1).$$

(b: even; $2^{2l}+1$: odd)

Chinese remainder theorem:

If $n_1, \dots, n_k$ are pairwise coprime, and if $a_1, \dots, a_k$ are any integers, then the system

$$\begin{cases} x \equiv a_1 \bmod n_1 \\ \vdots \\ x \equiv a_k \bmod n_k \end{cases}$$

(N) $n_1 n_2 \cdots n_k$

$x = a \bmod N$

has a unique solution modulo $(n_1 \cdots n_k)$.

Using this first compute $z_j \pmod{2^{2l}+1}$.

$$w^{b/2} + 1 = 2^{2l}+1 \Rightarrow w = 2^{\frac{4l}{b}}.$$

$O(b^2 \log b \cdot \log w)$
$+ O(b \cdot M(2l))$

Second, we compute $Z_j \pmod b$

$(A_0, \dots, A_{b-1}) *_- (B_0, \dots, B_{b-1}) \pmod b$.

$(A_0', \dots, A_{b-1}') *_- (B_0', \dots, B_{b-1}') \pmod b$

$\left.\begin{array}{l} A_i' = A \pmod b \\ B_i' = B \pmod b. \end{array}\right\}$

Let $A'' = A_{b-1}'' \cdots A_0''$ and $B'' = B_{b-1}'' \cdots B_0''$ where

$A_i'' = \underbrace{0000\cdots0}_{2\log b} A_i'$

$B_i'' = \underbrace{0\cdots\cdots0}_{2\log b} B_i'$.

$\left.\begin{array}{l} A_i'' \\ B_i'' \end{array}\right\}$ total of $3\log b$ bits.

Consider $A'' \times B''$. $\left.\begin{array}{l} \\ \\ \end{array}\right\}$ No. of bit size $3b\log b$. Apply Karatsuba $\to O((b\log b)^{\log\frac{3}{2}})$

↑ claim: No "overflow" happens.

↳ Total size of any element in the product has at most $3\log b$ bits.

Put both of them together to get Z_j.

$N = n_1 n_2$

Bezout's identity:

$\exists\, m_1, m_2$ s.t

$m_1 n_1 + m_2 n_2 = \gcd(n_1, n_2) = 1$.

$m_1 n_1 = 1 \pmod{n_2}$

$x \equiv a_1 \pmod{n_1}$

$x \equiv a_2 \pmod{n_2}$

$\left.\begin{array}{l} m_1 n_1 + m_2 n_2 = 1. \\ m_1 = 1 \\ m_2 = 2^{2l+1} - \frac{2^l}{b}. \end{array}\right\}$

$x = a_1 m_2 n_2 + a_2 m_1 n_1$

$= a_1 + (a_2 - a_1) m_1 n_1 \pmod{n_1 n_2}$

$m_1 = n_1^{-1} \bmod n_2$

$m_2 = n_2^{-1} \bmod n_1$

$$T(n) = a\,T\left(\frac{n}{b}\right) + O(n)$$

$$= n^{\log_b a} = n^{\frac{\log a}{\log b}} \leftarrow$$

Toom-Cook Knuth.