Fast Polynomials Multiplication Using FFT

Li Chen

lichen.xd at gmail.com

Xidian University

January 17, 2014

Outline

References

[1] Doz Dr A Schönhage and Volker Strassen. Schnelle multiplikation grosser zahlen. *Computing*, 7(3-4):281–292, 1971.

[2] Martin Fürer. Faster integer multiplication. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 57–66. ACM, 2007.

[3] Martin Fürer. Faster integer multiplication. *SIAM Journal on Computing*, 39(3):979–1005, 2009.

[4] Anindya De, Piyush P Kurur, Chandan Saha, and Ramprasad Saptharishi. Fast integer multiplication using modular arithmetic. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 499–506. ACM, 2008.

[5] Anindya De, Piyush P Kurur, Chandan Saha, and Ramprasad Saptharishi. Fast integer multiplication using modular arithmetic. *SIAM Journal on Computing*, 42(2):685–699, 2013.

Claim: Our slides are based on reference [1], [2], [3], [4], [5].

1 Discrete Fourier Transform (DFT)

Definition 1.1 (Discrete Fourier Transform (DFT)) Let $X = (x_0, x_1, \cdots, x_{N-1})$ be a N-length sequence, the *discrete fourier transform* of X is defined as a N-length sequences $F(X) = (f_1, f_2, \cdots, f_N)$, where

$$f_k = \sum_{n=0}^{N-1} x_n \omega_N^{nk}, \ k = 0, 1, \cdots, N-1.$$

where ω_N is the a principal Nth root of unity in a ring R (with unity).

Note1: Let R is a ring with unity, $\alpha \in R$ is called a principal Nth root of unity, if

$$\alpha^N = 1$$

and

$$\sum_{n=0}^{N-1} \alpha^{kn} = 0,\ 1 \leq k < N. \tag{1}$$

Note2: If R is a integral domain, it is sufficient to choose α as a primitive Nth root of unity, which replaces the condition (1) by

$$\alpha^k \neq 1,\ 1 \leq k < N. \tag{2}$$

Proof: Take $\beta = \alpha^k,\ 1 \leq k < N$, since $\alpha^N = 1,\ \beta^N = \alpha^{kN} = 1$, thus we have $\beta^N - 1 = (\beta - 1)\sum_{n=1}^{N-1} \beta^n = 0$, which implies $\sum_{n=1}^{N-1} \beta^n = 0$.

Theorem 1.2 (Inverse of Discrete Fourier Transform (IDFT)) Let $X = (x_0, x_1, \cdots, x_{N-1})$ be a N-length sequence, $F(X) = (f_1, f_2, \cdots, f_N)$ is the discrete fourier transform of X, then

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} f_k \omega_N^{-kn},\ k = 0, 1, \cdots, N-1.$$

where ω_N is the a principal N-th root of unity in a ring R, and $\frac{1}{N}$ is multiplicative inverse of N in R (if this inverse does not exist, the DFT cannot be inverted).

Proof

$$
\begin{aligned}
\frac{1}{N}\sum_{k=0}^{N-1} f_k \omega_N^{-kn} &= \frac{1}{N}\sum_{k=0}^{N-1}\Big(\sum_{j=0}^{N-1} x_j \omega_N^{jk}\Big)\omega_N^{-kn} \\
&= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1} x_j \omega_N^{(j-n)k} \\
&= \frac{1}{N}\sum_{j=0}^{N-1} x_j \sum_{k=0}^{N-1} \omega_N^{(j-n)k} \\
&= \frac{1}{N} x_n \sum_{k=0}^{N-1} \omega_N^{0} + \frac{1}{N}\sum_{j\neq n} x_j \sum_{k=0}^{N-1} \omega_N^{(j-n)k} \\
&= x_n + 0
\end{aligned}
$$

$\square$

2 Discrete Convolution

Definition 2.1 (Acyclic Convolution) Let $X = (x_0, x_1, \cdots, x_{N-1}), Y = (y_0, y_1, \cdots, y_{N-1})$ be two N-length sequences, then the *acyclic convolution* or *linear convolution* is defined as a $2N+1$-length sequences $Z = (z_0, z_1, \cdots, z_{2(N-1)})$, where

$$z_i = \sum_{\substack{j \in [0, N-1] \\ k \in [0, N-1]}}^{j+k=i} x_j y_k, \ i = 0, 1, \cdots, 2(N-1).$$

And the *cyclic convolution* or *wrapped convolution* is defined as a N-length sequences $\tilde{Z} = (\tilde{z}_0, \tilde{z}_1, \cdots, \tilde{z}_{N-1})$, where

$$\tilde{z}_i = z_i + z_{i+N}, \ i = 0, 1, \cdots, N-1, \ (\text{pad } z_{2N-1} = 0).$$

or equivalent,

$$\tilde{z}_i = \sum_{j=0}^{N-1} x_j y_{i-j}, \ i = 0, 1, \cdots, N-1, \ (y_{-k} = y_{N-k}, \ k = 0, 1, \cdots, N-1).$$

Definition 2.2 (Negacyclic Convolution) Let $X = (x_0, \cdots, x_{N-1}), Y = (y_0, \cdots, y_{N-1})$ be two N-length sequences, then the *negacyclic convolution* is defined as a N-length sequences $\tilde{Z} = (\tilde{z}_0, \tilde{z}_1, \cdots, \tilde{z}_{N-1})$, where

$$\tilde{z}_i = z_i - z_{i+N},\ i = 0, 1, \cdots, N-1,\ (\text{pad } z_{2N-1} = 0).$$

or equivalent,

$$\tilde{z}_i = \sum_{j=0}^{N-1} x_j y_{i-j},\ i = 0, 1, \cdots, N-1,\ (y_{-k} = -y_{N-k},\ k = 0, 1, \cdots, N-1).$$

Theorem 2.3 (Convolution Theorem) Let $X = (x_0, \cdots, x_{N-1})$, $Y = (y_0, \cdots, y_{N-1})$ be two N-length sequences, take $F(\cdots)$ as the discrete fourier transform on a N-length sequences $(\cdots)$, then

$$F(X * Y) = N \cdot F(X) \otimes F(Y)$$

where $*$ indicates cyclic convolution, and $\otimes$ indicates component-wise multiplication.

Proof

Let $\tilde{z}_0, \cdots, \tilde{z}_{N-1}$ be the cyclic convolution of X, Y, i.e.,

$$\tilde{z}_j = \sum_{j=0}^{N-1} x_j y_{n-j}$$

and let $F(x * Y) = (f_0, f_1, \cdots, f_{N-1})$. For each $k \in [0, N-1]$,

$$\begin{aligned} f_k = \sum_{n=0}^{N-1} \tilde{z}_n \omega_N^{nk} &= \sum_{n=0}^{N-1} \Big(\sum_{j=0}^{N-1} x_j y_{n-j} \Big) \omega_N^{nk} \\ &= \sum_{n=0}^{N-1} \Big(\sum_{j=0}^{N-1} x_j y_{n-j} \Big) \omega_N^{jk+(n-j)k} \\ &= \sum_{n=0}^{N-1} \Big(\sum_{j=0}^{N-1} x_j \omega_N^{jk} \cdot y_{n-j} \omega_N^{(n-j)k} \Big) \\ &= \sum_{j=0}^{N-1} x_j \omega_N^{jk} \cdot \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} x_j y_{n-j} \omega_N^{(n-j)k} \\ &= N \cdot \sum_{j=0}^{N-1} x_j \omega_N^{jk} \cdot \sum_{j=0}^{N-1} y_j \omega_N^{jk} \end{aligned}$$

$\square$

Lemma 2.4 Let $X = (x_0, \cdots, x_{N-1})$, $Y = (y_0, \cdots, y_{N-1})$ be two N-length sequences, let X' be a $2N-1$-length sequences $(x_0, \cdots, x_{N-1}, 0, \cdots, 0)$, i.e. padding X with 0, similarly, let Y' be a $2N-1$-length sequences $(y_0, \cdots, y_{N-1}, 0, \cdots, 0)$, i.e. padding Y with 0, then

$$\mathsf{AcyclicConvolution}(X, Y) = \mathsf{CyclicConvolution}(X', Y')$$

Proof

□

3 Fast Fourier Transform (FFT)

Cooley-Tukey Algorithm (radix-2)

Let $X = (x_0, \cdots, x_{N-1})$ be a sequence with length $N = 2M$, and $F(X) = (f_0, \cdots, f_{N-1})$ be the discrete fourier transform of X, i.e.

$$\begin{aligned} f_k &= \sum_{n=0}^{N-1} x_n \omega_N^{nk} \\ &= \sum_{m=0}^{M-1} x_{2m} \omega_N^{2mk} + \sum_{m=0}^{M-1} x_{2m+1} \omega_N^{(2m+1)k} \\ &= \sum_{m=0}^{M-1} x_{2m} \omega_{\frac{N}{2}}^{mk} + \omega_N^k \sum_{m=0}^{M-1} x_{2m+1} \omega_{\frac{N}{2}}^{mk} \end{aligned}$$

Note: ω is a principle Nth root of unity $\Rightarrow \omega^2$ is a principle $\frac{N}{2}$th root of unity.

Thus, let $E_k = \sum_{m=0}^{M-1} x_{2m}\omega_{\frac{N}{2}}^{mk}$, and $D_k = \sum_{m=0}^{M-1} x_{2m+1}\omega_{\frac{N}{2}}^{mk}$, we can write $f_k = E_k + \omega_N^k D_k$.

Since $E_{k+\frac{N}{2}} = E_k, D_{k+\frac{N}{2}} = D_k, k = 0, \cdots, M-1$, and $\omega_N^{k+\frac{N}{2}} = -\omega_N^k$, finally we have, for $0 \le k \le M-1$

$$f_k = E_k + \omega_N^k D_k$$
$$f_{k+\frac{N}{2}} = E_k - \omega_N^k D_k$$

And Now we can view E_k as the kth term of the discrete fourier transform on the $\frac{N}{2}$-length sequence $(x_0, x_2, \cdots, x_{N-2})$, and view D_k as the kth term of the discrete fourier transform on the $\frac{N}{2}$-length sequence $(x_1, x_3, \cdots, x_{N-1})$.

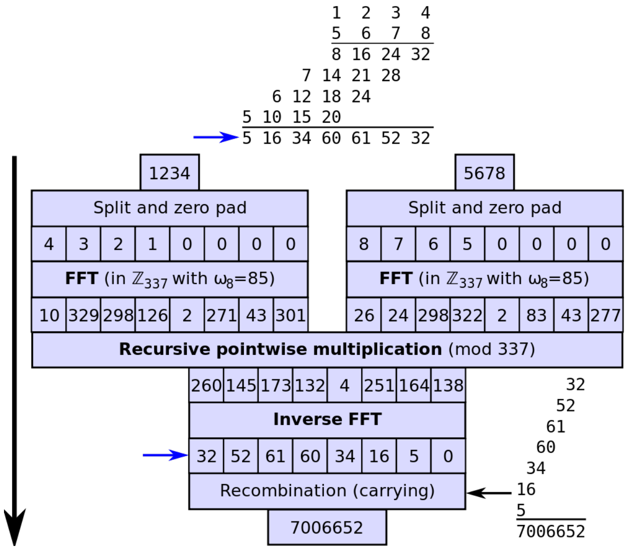
Suppose $N = 2^m$, by reduction, the asymptotic complexity of FFT is:

$$O(N \log N)$$

4 Number Theoretic Transform (NTT)

The number theoretic transform (NTT) is obtained by specializing the discrete Fourier transform on a special ring Z_p, the integers modulo a prime p, which is a finite field.

Lemma 4.1 Given a prime integer p, there exist a primitive nth root of unity in Z_p if and only if $n|p-1$.


1 2 3 4
5 6 7 8
8 16 24 32
7 14 21 28
6 12 18 24
5 10 15 20
5 16 34 60 61 52 32
1234
5678
Split and zero pad
4 3 2 1 0 0 0 0
8 7 6 5 0 0 0 0
FFT (in $\mathbb{Z}_{337}$ with ω_8=85)
10 329 298 126 2 271 43 301
26 24 298 322 2 83 43 277
Recursive pointwise multiplication (mod 337)
260 145 173 132 4 251 164 138
Inverse FFT
32 52 61 60 34 16 5 0
Recombination (carrying)
7006652
32
52
61
60
34
16
5
7006652

5 More Optimize & Application Scenario

Internal Discussed...

Thanks! & Questions?