高速乗算法の設計と実装 (5)

梅谷 武

作成：2001-09-24 更新：2005-04-20

高速乗算法 (5) は係数を 2 語と 1 語に分解する方法である。
IMS:20010924001; NDC:412.1; keywords:高速乗算法;

目　　次

1　係数の分解

1.1　数列 T(n)

高速乗算法 (1)(→ [F3]) で導入した $T(n)$ を再度使用する。

$$T(n) = 2^{2t} - 2^t + 1, t = 2^n, n > 0$$

とおくと

$$2^{2K} - 2^{K+1} + 1 < T(k), K = 2^k$$

が成り立つ。$T(n), n = 0, 1, 2, 3, 4, 5, 6$ を素数判定すると次のようになる。

$T(0) = 2(2-1)+1 = 3$: 素数
$T(1) = 2^2(2^2-1)+1 = 13$: 素数
$T(2) = 2^4(2^4-1)+1 = 241$: 素数
$T(3) = 2^8(2^8-1)+1 = 65281 = 97 * 673$: 合成数
$T(4) = 2^{16}(2^{16}-1)+1 = 4294901761 = 193 * 22253377$: 合成数
$T(5) = 2^{32}(2^{32}-1)+1 = 18446744069414584321$: 素数
$T(6) = 2^{64}(2^{64}-1)+1$: 合成数

$T(5)$ が素数になるのでこれを使用する。$T(5)$ を法とする整数の剰余環 $\mathbf{Z}_{T(5)}$ は有限体となるが、この元は 1 語を 32bit とするとき 2 語で表現できる。この積は

$$A2^{64} + B, 0 \leqq A, B \leqq 2^{64} - 1$$

と 128bit 長で表現されるが、このとき、

$$\begin{aligned} A2^{64} + B &= (2^{64} - 2^{32} + 1)A + (2^{32} - 1)A + B \\ &\equiv (2^{32} - 1)A + B(mod T(5)) \end{aligned}$$

より除算を使わずに剰余演算ができる。

1.2 T(5) を使った離散フーリエ変換の型

$T(5)$ は素数であり、$\mathbf{F}_{T(5)}$ 上に長さ N の離散フーリエ変換が存在するための必要十分条件は、

$$N | 2^{32}(2^{32} - 1)$$

であるが、

$$N = 2^n, 0 < n \leqq 30$$

はこの条件を満たし、さらにこのとき $T(5)$ の最小原始根 7 を使って $\mathbf{F}_{T(5)}$ 上の 1 の原始 N 乗根を

$$7^t, t = 2^{32-n}(2^{32} - 1), 0 < n \leqq 30$$

と表現することができる。

1.3 数列 V(n)

$T(5)$ によって係数の評価式

$$c_r \leqq N(2^{32} - 1)^2 = N(2^{64} - 2^{33} + 1)$$

の

$$2^{64} - 2^{33} + 1$$

を 2 語で表現できる素数によって上から押さえることができた。次に

$$N = 2^n, 0 < n \leqq 30$$

を 1 語で表現できる素数によって上から押さえることを考える。

$$V(n) = 2^n(2^{32-n} - 1) + 1, 0 < n < 32$$

とおくとこれは 1 語で表現できる。この中で素数となるものは、

$$\begin{aligned} V(20) &= 2^{20}(2^{12} - 1) + 1 = 4293918721 \\ V(30) &= 2^{30}(2^2 - 1) + 1 = 3221225473 \end{aligned}$$

の 2 つである。離散フーリエ変換の長さをなるべく長くするために $V(30)$ を使う。

$$2^{31} < V(30) = 2^{32} - 2^{30} + 1$$

が成り立っているので、

$$N = 2^n < V(30), 0 < n \leqq 30$$

である。$V(30)$ を法とする整数の剰余環 $\mathbf{Z}_{V(30)}$ は有限体となるが、この元は 1 語で表現できる。この積は

$$A2^{32}+B, 0 \leqq A, B \leqq 2^{32}-1$$

と 64bit 長で表現されるが、このとき、

$$\begin{aligned} A2^{32}+B &= (2^{32}-2^{30}+1)A+(2^{30}-1)A+B \\ &\equiv (2^{30}-1)A+B(mod\, V(30)) \end{aligned}$$

より除算を使わずに剰余演算ができる。

1.4 V(30) を使った離散フーリエ変換の型

$V(30)$ は素数であり、$\mathbf{F}_{V(30)}$ 上に長さ N の離散フーリエ変換が存在するための必要十分条件は、

$$N | 2^{30}(2^2-1)$$

であるが、

$$N = 2^n, 0 < n \leqq 30$$

はこの条件を満たし、さらにこのとき $V(30)$ の最小原始根 5 を使って $\mathbf{F}_{V(30)}$ 上の 1 の原始 N 乗根を

$$5^t, t = 2^{30-n}(2^2-1), 0 < n \leqq 30$$

と表現することができる。

2 高速乗算法 (5)

2.1 高速乗算法 (5) の構造

Step 1. 数の P 進表現

基数を $P = 2^{32}$ とし、長さを

$$N = 2^n, 0 < n \leqq 32$$

とする。正の整数 a, b が次のように P 進表現されるものとする。

$$\begin{aligned} a &= a_{N-1}P^{N-1}+\cdots+a_1P+a_0, 0 \leqq a_i < P \\ b &= b_{N-1}P^{N-1}+\cdots+b_1P+b_0, 0 \leqq b_i < P \end{aligned}$$

この積について $0 \leqq ab < P^N$ が成り立つと仮定する。

Step 2. 多項式としての積の計算

$P^K \equiv 1 (\mathrm{mod}\, P^N-1)$ より、

$$c = ab \equiv \sum_{r=0}^{N-1} \Big(\sum_{s+t \equiv r(modN)} a_s b_t \Big) P^r (\mathrm{mod}\, P^N-1)$$

となる。この係数 $c_r = \sum_{s+t \equiv r(modN)} a_s b_t$ の大きさを評価すると、

$$0 \leqq c_r \leqq N(2^{32}-1)^2 < U(5) = 2^{32}(2^{64}-1)+1$$

となる。

Step 3. **離散フーリエ変換による係数の計算**

c_r は、$(N, \mathbf{F}_{U(5)}, \zeta), \zeta = 11^t, t = 2^{32-n}(2^{64}-1)$ 型の離散 Fourier 変換を利用して次のように計算する。

$$F(a)_k = \sum_{s=0}^{N-1} a_s \zeta^{sk} \quad (mod\, U(5))$$

$$F(b)_k = \sum_{t=0}^{N-1} b_t \zeta^{tk} \quad (mod\, U(5))$$

$$c_r = N^{-1} \sum_{k=0}^{N-1} F(a)_k F(b)_k \zeta^{-kr} \quad (mod\, U(5))$$

Step 4. **桁上げ処理**

最後に $c_r, r = 0, \cdots, K-1$ に桁上げ処理を施すことで、$c = ab$ の P 進表現が得られる。

参考文献

高速乗算法

[F1] 梅谷 武, 離散 *Fourier* 変換

[F2] 梅谷 武, ストラッセン-ショーンハーゲ法

[F3] 梅谷 武, 高速乗算法の設計と実装 *(1)*

[F4] 梅谷 武, 高速乗算法の設計と実装 *(2)*

[F5] 梅谷 武, 高速乗算法の設計と実装 *(3)*