# 高速乗算法の設計と実装 (4)

梅谷 武

作成：2001-09-20 更新：2005-04-20

高速乗算法 (3) において係数を分解しない方法の実験を行なったが、剰余演算の語長が 6 語であったため、2 語と 4 語に分解した高速乗算法　よりかなり計算量が多くなった。高速乗算法　は係数を分解しないで剰余演算の語長を 3 語にする。
IMS:20010920002; NDC:412.1; keywords:高速乗算法;

目　　次



## 1　高速乗算法 (4) の設計

### 1.1　数列 U(n)

係数の評価式

$$c_r \leqq N(2^K-1)^2 = N(2^{2K}-2^{K+1}+1)$$
$$K = 2^5 = 32$$
$$N = 2^n, 0 < n \leqq 32$$

から

$$N(2^{2K}-2^{K+1}+1) \leqq p$$

なる素数 $p$ であって、$\mathbf{F}_p$ の演算、すなわち法 $p$ の剰余演算が高速に計算できるものを探す。

$$U(n) = 2^{3t}-2^t+1, t = 2^n, n > 0$$

とおくと

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

が成り立つ。$U(n), n = 0, 1, 2, 3, 4, 5, 6$ を素数判定してみる。

$$U(0) = 2(2^2 - 1) + 1 = 7 : \text{素数}$$
$$U(1) = 2^2(2^4 - 1) + 1 = 61 : \text{素数}$$
$$U(2) = 2^4(2^8 - 1) + 1 = 4081 : \text{合成数}$$
$$U(3) = 2^8(2^{16} - 1) + 1 = 16776961 : \text{素数}$$
$$U(4) = 2^{16}(2^{32} - 1) + 1 = 281474976645121 : \text{合成数}$$
$$U(5) = 2^{32}(2^{64} - 1) + 1 : \text{素数}$$
$$U(6) = 2^{64}(2^{128} - 1) + 1 : \text{合成数}$$

$U(5)$、すなわち $K = 2^5 = 32$ の場合に素数になっているが、

$$A2^{96} + B, 0 \leqq A, B \leqq 2^{96} - 1$$

と表現するとき、

$$\begin{aligned} A2^{96} + B &= (2^{96} - 2^{32} + 1)A + (2^{32} - 1)A + B \\ &\equiv (2^{32} - 1)A + B (mod U(5)) \end{aligned}$$

となり除算を使わずに 3 語の剰余演算ができる。

### 1.2 離散フーリエ変換の型

$p = U(5) = 2^{32}(2^{64} - 1) + 1$ とする。これは素数であり、

$$c_r < p$$

という係数の評価式が成り立つ。さらに $\mathbf{F}_p$ 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、

$$N | 2^{32}(2^{64} - 1)$$

であるが、

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

はこの条件を満たし、さらにこのとき $p$ の最小原始根 11 を使って $\mathbf{F}_p$ 上の 1 の原始 $N$ 乗根を

$$11^t, t = 2^{32-n}(2^{64} - 1), 0 < n \leqq 32$$

と表現することができる。

## 2 高速乗算法 (4)

### 2.1 高速乗算法 (4) 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。

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

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、

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

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)*