

2.3 乗算

2.3.1 多項式の乗算

例
$$\begin{cases} a(x) = a_3x^3 + a_2x^2 + a_1x + a_0, & a_j \in \mathbb{Z}, a_3 \neq 0. \\ b(x) = b_1x + b_0, & b_j \in \mathbb{Z}, b_1 \neq 0. \end{cases}$$

$$\begin{array}{r} \begin{array}{cccc} a_3 & a_2 & a_1 & a_0 \\ x) & & b_1 & b_0 \\ \hline a_3b_0 & a_2b_0 & a_1b_0 & a_0b_0 \\ a_3b_1 & a_2b_1 & a_1b_1 & a_0b_1 \\ \hline c_4 & c_3 & c_2 & c_1 & c_0 \end{array} \end{array}$$

Algorithm (1変数多項式の乗算)

入力: $a(x) = \sum_{i=0}^k a_i x^i$, $b(x) = \sum_{i=0}^m b_i x^i \in \mathbb{Z}[x]$;

出力: $c(x) = \sum_{i=0}^{k+m} c_i x^i \in \mathbb{Z}[x]$ s.t. $c(x) = a(x) \cdot b(x)$.

(1) for $i \in [0..k+m]$ do $c_i \leftarrow 0$; (初期化)

(2) for $i \in [0..k]$ do
 for $j \in [0..m]$ do
 $c_{i+j} \leftarrow c_{i+j} + a_i b_j$;

(3) return $(c_0, c_1, \dots, c_{k+m})$;

⑧ 演算回数、見積もり、(各ステップ毎)

(1) ループ: $k+m+1$ 回, 代入は無視.

(2) i のループ: $k+1$ 回.

j のループ: $m+1$ 回.

乗算 1回, 加算 1回.

(3) return は無視.

$\therefore$ 合計 $2(k+1)(m+1) = \Theta(km)$.

★ 演算回数は $a(x)$ と $b(x)$ の次数の積に比例.

(しかし、ワード単位の計算量の見積もりは、次の通り
 多倍長整数の乗算の見積もりが必要.)

2.3.2 多倍長整数と単精度整数の乗算

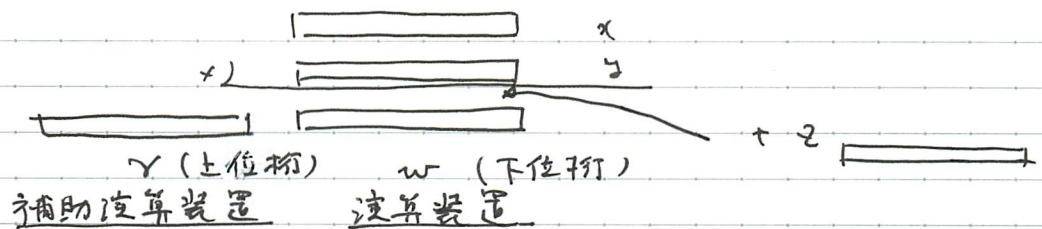
① 単精度整数としての乗算 (ワード = m ビット)

$$a, b \in \mathbb{Z}$$

$$\begin{array}{ccc} a & \times & b \\ n \text{ 桁} & & n \text{ 桁} \end{array} \longrightarrow \begin{array}{c} ab \\ 2n \text{ 桁} \end{array}$$

ワード ワード $\longrightarrow$ ワード. 分の記憶領域が必要.

そこで.



で以下と扱う. これを $[w \ y] \leftarrow xy + z$ で表す.

$$\begin{array}{c} \parallel \\ y \cdot 2^n + w \end{array}$$

② 多倍長整数と単精度整数の乗算のアルゴリズム

Algorithm (多倍長整数と単精度整数の乗算)

入力: 多倍長整数 $a = [a_0 a_1 \dots a_{k-1}]$,
 $a_j \in [0..2^n - 1]$ for $j \in [0..k-1]$.
 単精度整数 $b \in [0..2^n - 1]$.

出力: 多倍長整数 $c = [c_0 c_1 \dots c_k]$,
 $c_j \in [0..2^n - 1]$ for $j \in [0..k]$
 s.t. $c = ab$.

(1) $\gamma_0 \leftarrow 0$;

(2) for $i \in [0..k-1]$ do $\gamma_{i+1} \leftarrow a_i \cdot b + \gamma_i$;

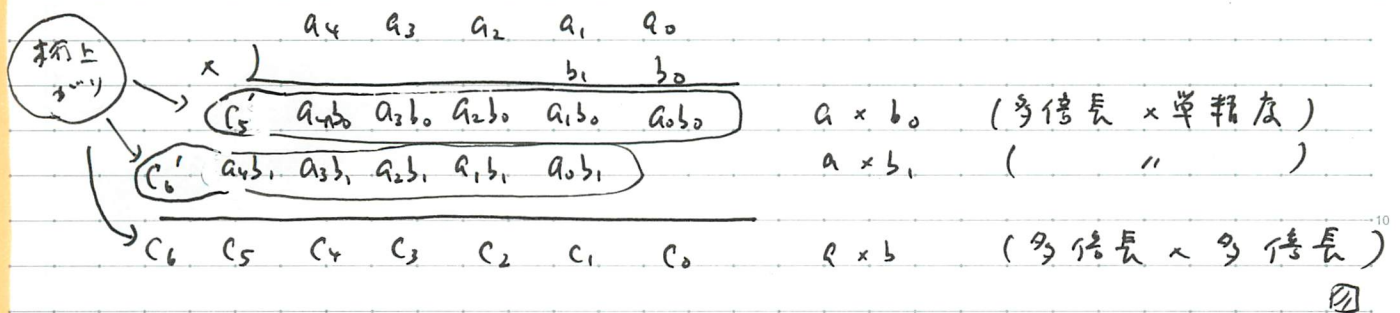
(3) $c_k \leftarrow \gamma_k$;

(4) return $c = [c_0 c_1 \dots c_k]$; 乗算 1 回 加算 1 回 ②

・ 計算量: $O(k) = O(\text{len}(a))$.

2.3.3 多倍長整数としての乗算

例) $a = [a_0, a_1, a_2, a_3, a_4]$, $b = [b_0, b_1]$ のとき.



Algorithm (多倍長整数としての乗算)

入力: 多倍長整数 $a = [a_0, a_1, \dots, a_{k-1}]$

$b = [b_0, b_1, \dots, b_{m-1}]$

出力: 多倍長整数 $c = [c_0, c_1, \dots, c_{k+m-1}]$

(1) $[c_0 \dots c_k] \leftarrow a \cdot b_0$; 乗算 $O(\text{len}(a))$

(2) for $v \in [1 \dots m-1]$ do $10^{-7} \cdot m-1$
 $[c_v \dots c_{k+v}] \leftarrow [c_v \dots c_{k+v-1}] + a \cdot b_v$

(3) return $[c_0, c_1, \dots, c_{k+m-1}]$; 加算 乗算
 $O(\text{len}(a)), O(\text{len}(a))$

計算量

$$O(\text{len}(a)) + (m-1) \cdot O(\text{len}(a))$$

$$= m \cdot O(\text{len}(a)) = O(\text{len}(a) \cdot \text{len}(b))$$

$$\text{len}(b)$$