NO. 26

DATE 6 · 7 · 2015

6/22

2.3 乗算.

2.3.1 多項式の乗算.

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. $c(x) = a(x)\,b(x)$.

(1) for $i \in [0..k+m]$ do $c_i \leftarrow 0$; ← ループ $k+m+1$回. 代入の無視.

(2) for $i \in [0..k]$ do ループ $k+1$回
  for $j \in [0..m]$ do ループ $m+1$回.
    $c_{i+j} \leftarrow c_{i+j} + a_i b_j$; 乗算1回. 加算1回

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

例 $\begin{cases} a(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0, \\ b(x) = b_1 x + b_0. \end{cases}$

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

□

$(k次) + (m次) \longrightarrow (k+m)$次.

◎ 演算回数の見積り.

以下、カッコ内の数字の上のアルゴリズムの各ステップとする.

(1) ループ $k+m+1$回. 代入は無視.

(2)外側ループ $k+1$回. (内)ループ $m+1$回.
乗算1回. 加算1回.

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は, 以下で述べる
多倍長整数の見積りが必要.

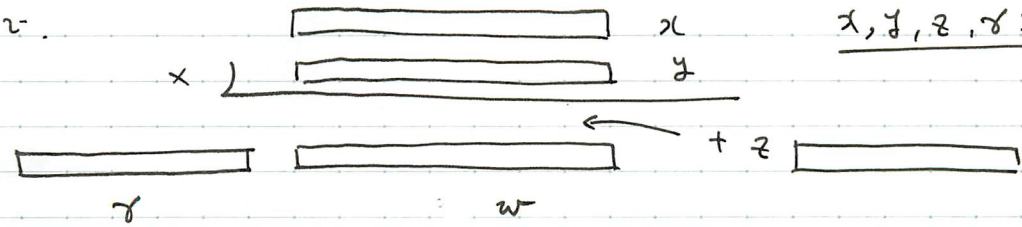
2.3.2 多倍長整数と単精度整数の乗算.

◎ 単精度整数どうしの乗算 (1ワード = nビット)

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

a	$\times$	b	$=$	ab
n桁		n桁	→	2n桁
1ワード		1ワード	→	2ワード. 分の記憶領域が必要.

ここで.



x, y, z, γ: レジスタ (1ワード)

(補助演算装置) (演算装置)

で データを扱う. これと

$$[w\ \gamma] \leftarrow xy + z$$

で表す.

◎ 多倍長整数と単精度整数の乗算のアルゴリズム

6-21-2015

Algorithm （多倍長整数と単精度整数の乗算）

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

$a_j \in [0..2^n-1]$ for $j \in [0..k-1]$

単精度整数　$b \in [0..2^n-1]$

出力　多倍長整数　$c = [c_0\ c_1\ \cdots\ c_k]$,

$c_j \in [0..2^n-1]$ for $j \in [0..k]$

s.t.　$c = ab$.

(1)　$\gamma_0 \leftarrow 0$;

(2)　for $i \in [0..k-1]$ do　← ループ k回

$[c_i\ \gamma_{i+1}] \leftarrow a_i b + \gamma_i$;

（$a_i b$：乗算1回，$+\gamma_i$：加算1回）

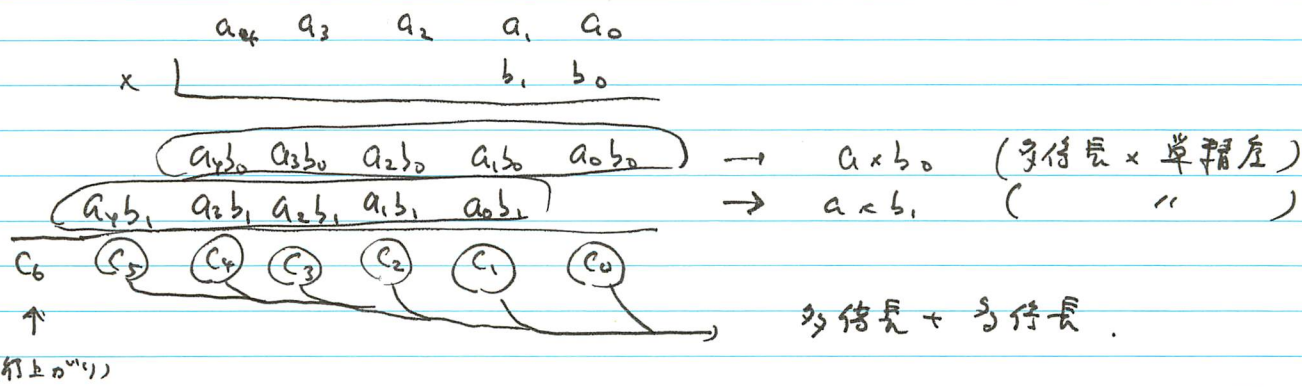
(3)　$c_k \leftarrow \gamma_k$;

(4)　return　$c = [c_0\ c_1\ \cdots\ c_k]$;

• 計算量：　$O(k) = O(\mathrm{len}(a))$.

2.3.3　多倍長整数どうしの乗算

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



Algorithm （多倍長整数どうしの乗算）

入力　多倍長整数　$a = [a_0\ a_1\ \cdots\ a_{k-1}]$
$b = [b_0\ b_1\ \cdots\ b_{m-1}]$

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

s.t. $c = ab$.

(1) $[c_0 \cdots c_k] \leftarrow ab_0$;　乗算 $O(len(a))$

(2) for $i \in [1..m-1]$ do　← ループ $m-1$ 回.

$[c_i \cdots c_{k+i}] \leftarrow [c_i \cdots c_{k+i-1}] + ab_i$)

加算 $O(len(a))$　乗算 $O(len(a))$

(3) return c;

□

・計算量：$O(len(a)) + (m-1) \cdot O(len(a))$

$= m \cdot O(len(a)) = O(len(a) \cdot len(b))$.

$m = len(b)$

2.4 剰余つき除算.

◎ 整数の剰余つき除算.

Thm. 1 (整数の剰余つき除算の性質).

$a, b \in \mathbb{Z}$, $b > 0$ のとき,

$\exists q, r \in \mathbb{Z}$ s.t. $a = bq + r$ かつ $0 \le r < b$.

Proof → 問題 2.13. (6/23) ↓ □

Def. 2 (整数の剰余つき除算) (6/29)

a, b, q, r as in Thm. 1.

- a, b に対し, 上の q, r を定める計算を a の b による除算といい, $a \div b$ で表す.
- a: 被除数 (dividend)
- b: 除数 (divisor)
- q: 商 (quotient) = quo(a, b)
- r: 剰余 (remainder) = $a \bmod b$.
- $a \div b$ に対し, q と r の組を同時に求めること

$$(q, r) \leftarrow a \div b$$

で表す. □

2.4.1 多項式の剰余つき除算.

$a(x), b(x) \in \mathbb{Z}[x]$, $b(x)$: monic

に対し, 多項式の剰余つき除算を Def. 2 と同様に定義する.

例) $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$, $b(x) = 1x + b_0$ の場合.

$$\begin{array}{r|l} & a_3 \\ \hline 1 \quad b_0 & a_3 \quad a_2 \quad a_1 \quad a_0 \\ & a_3 \quad a_3b_0 \\ \hline & a_2 - a_3b_0 \quad \cdots \end{array}$$

□