Polynomial multiplication and division using a Heap

September 17, 2021 10:10 AM

Multiplying polynomials using a heap. Johnson 1974.

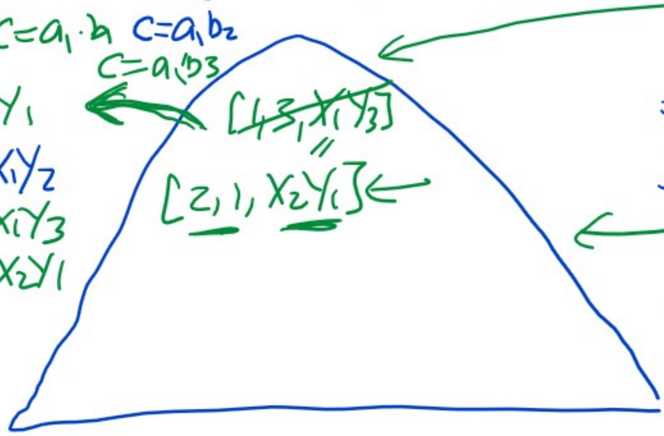
Let $f = f_1 + f_2 + \cdots + f_m = a_1 X_1 + a_2 X_2 + \cdots + a_m X_m$ s.t. $X_1 > X_2 > \cdots > X_m$.

$g = g_1 + g_2 + \cdots + g_l = b_1 Y_1 + b_2 Y_2 + \cdots + b_l Y_l$ s.t. $Y_1 > Y_2 > \cdots > Y_l$.

Let $h = f \cdot g = f_1 g + f_2 g + f_3 g + \cdots + f_m g$.

$c = a_1 \cdot b_1$ $c = a_1 b_2$ $c = a_1 b_3$

$h = c X_1 Y_1$
$+ a_1 b_2 X_1 Y_2$
$+ (a_1 b_3 X_1 Y_3$
$a_2 b_1) X_2 Y_1$



h
$= f_1 g = f_1 g_1 + f_1 g_2 + f_1 g_3 + \cdots + f_1 g_l$
$+ f_2 g = f_2 g_1 + f_2 g_2 + \cdots + f_2 g_l$
$+ f_3 g = f_3 g_1 + f_3 g_2 + \cdots + f_3 g_l$
$+ \cdots$
$+ f_m g = f_m g_1 + \cdots + f_m g_l$

(1) Create a heap H and insert $[1, 1, X_1 \cdot Y_1]$ into H.

$f_1 \cdot g_1$

$h := 0;$

(2) while $H \neq \{\}$ do
  extract $H_1 = [i, j, Z]$ from H and set $c = a_i \cdot b_j$.
  if $j = 1$ and $i < m$ insert $[i+1, 1, X_i Y_1]$ into H.
  if $j < l$ insert$([i, j+1, X_i Y_{j+1}])$ into H
  while $H \neq \{\}$ and $H_i = [-, -, Z]$ do
    extract $[i, j, Z]$ from H
    $c = c + a_i \cdot b_j$
    [update heap]
  end while
  if $c \neq 0$ then append $[c, Z]$ to h. // $h = h + c \cdot Z$.
end while
output h.

Coefficient arithmetic: $c = \left(\sum a_i \cdot b_j \right)$ can be done with

Coefficient arithmetic: $C = \boxed{\sum_{i,j} a_i \cdot b_j}$ can be done with ($a_i, b_j \in \mathbb{Z}$) two accumulators

a_i b_j C $0 \quad 00$ $a_i \cdot b_j$ h

to eliminate $O(m \cdot n)$ pieces of garbage.

How many monomial comparisons does heap X do in the worst case?

Every term in the product $[i,j,\ X_i \cdot Y_j]$ is inserted into the heap eventually, then is extracted, eventually.

$$\#\text{comparisons} \le m \cdot \ell \cdot \text{Cost}(\text{insert} + \text{extract}).$$

$$|H| \le m \qquad \le m\ell\,[O(\log m) + O(\log m)]$$

$$= O(m\ell \log m).$$

If $\#f > \#g$ we should multiply $g \times f$. Then

$$\#\text{comparisons} \le O(m\ell \underline{\log \min(m,\ell)})$$

$$(f = a_1X_1 + a_2X_2)(g = b_1Y_1 + b_2Y_2 + \cdots)$$

Note this is more expensive than repeated merging in the dense case: $O(m\ell)$ comparisons.

Division. Test if $g | f$ and if so output q s.t. $f = gq$.

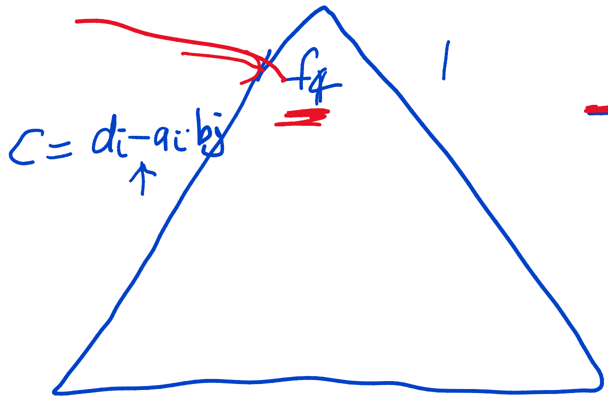
Let $q = q_1 + q_2 + \cdots + q_{\#q} = C_1Z_1 + \cdots + C_{\#q}Z_{\#q}$.

If $g | f$ we will compute

$$f - q_1g - q_2g - \cdots - q_{\#}g = 0.$$

$$= f - \underline{(q_1 + q_2 + \cdots)g}$$

$$= f - q \cdot g \qquad \uparrow \text{heap.}$$



$$f = f_1 + f_2 + \cdots + f_m$$

$$- \left[\begin{array}{l} (q_1 g = \cancel{q_1 g_1} + q_1 g_2 + \cdots + q_1 g_\ell) \\ (q_2 g = \cancel{q_2 g_1} + q_2 g_2 + \cdots + q_2 g_\ell) \\ \quad \vdots \\ q_{\#q} g = q_{\#q} g_1 + \cdots + q_{\#q} g_\ell \end{array}\right]$$

Johnson's idea: use a heap to subtract
$q_1 g + q_1 g + \cdots + q_{\#q} g$ from f.

$$\#\text{ comparisons} \le \#q \cdot \#g \, O(\log(\#q+1))$$

$$= O(\#q \, \#g \, \underline{\log \#q}).$$

Can we get $O(\#q \, \#g \log \min(\#q, \#g))$?

Pearce & MBM: Yes.