Polynomial Division and Greatest Common Divisors

Com S 477/577

Sep 2, 2003

Let $u(x)$ and $v(x)$ be two polynomials such that $v(x) \neq 0$ and $\deg(u) \geq \deg(v)$. Suppose all the coefficients are real (or rational). Then there exists a *quotient* polynomial $q(x)$ and a *remainder* polynomial $r(x)$ such that

$$u(x) = q(x)v(x) + r(x), \qquad \deg(r) < \deg(v). \tag{1}$$

It is easy to see that there is at most one pair of polynomials $(q(x), r(x))$ satisfying (1); for if $(q_1(x), r_1(x))$ and $(q_2(x), r_2(x))$ both satisfy the relation with respect to the same polynomial $u(x)$ and $v(x)$, then $q_1(x)v(x) + r_1(x) = q_2(x)v(x) + r_2(x)$, so $(q_1(x) - q_2(x))v_{(}x) = r_2(x) - r_1(x)$. Now if $q_1(x) - q_2(x)$ is nonzero, we have $\deg((q_1 - q_2) \cdot v) = \deg(q_1 - q_2) + \deg(v) \geq \deg(v) > \deg(r_2 - r_1)$, a contradiction; hence $q_1(x) - q_2(x) = 0$ and $r_1(x) = r_2(x)$.

Given its uniqueness, we denote $q(x) = \lfloor \frac{u(x)}{v(x)} \rfloor$, analogous to the quotient in integer division. Obviously, $r(x) = u(x) - v(x)\lfloor \frac{u(x)}{v(x)} \rfloor$.

Let

$$\begin{aligned} u(x) &= u_m x^m + \cdots + u_1 x + u_0, \\ v(x) &= v_n x^n + \cdots + v_1 x + v_0, \end{aligned}$$

where $v_n \neq 0$ and $m \geq n \geq 0$, the following procedure finds the polynomials

$$\begin{aligned} q(x) &= q_{m-n} x^{m-n} + \cdots + q_0, \\ r(x) &= r_{n-1} x^{n-1} + \cdots + r_0 \end{aligned}$$

that satisfy (1).

Polynomial-Divide$\Big(u(x), v(x)\Big)$

1 $m \leftarrow \deg(u)$
2 $n \leftarrow \deg(v)$
3 **for** $k = m - n$ **downto** 0
4 $\quad q_k \leftarrow u_{n+k}/v_n$
5 $\quad$ **for** $j = n + k - 1$ **downto** k
6 $\qquad u_j \leftarrow u_j - q_k v_{j-k}$
7 $(r_{n-1}, \ldots, r_0) \leftarrow (u_{n-1}, \ldots, u_0)$

For example, let $u(x) = 3x^3 - 5x^2 + 10x + 8$ and $v(x) = x^2 + 2x - 3$. Then the **for** loop of lines 3–6 goes through two iterations and yields $q(x) = 3x - 11$ and $r(x) = 41x - 25$.

It is not difficult to see that the number of arithmetic operations involved in polynomial division is $O((m-n+1)n)$ if the procedure POLYNOMIAL-DIVIDE is used. In the next section, we will describe an algorithm that computes the quotient $\lfloor \frac{u(x)}{v(x)} \rfloor$ in time $O(n \lg n)$ if m is on the order n. Later on we will introduce a fast algorithm that computes the greatest common divisor of $u(x)$ and $v(x)$.

1 A Fast Division Algorithm

Let $u(x) = u_m x^m + \cdots + u_1 x + u_0$ and $v(x) = v_n x^n + \cdots + v_1 x + v_0$ be two polynomials of degrees m and n, respectively. Suppose we are to compute $q(x) = \lfloor \frac{u(x)}{v(x)} \rfloor$. First, let us transform the division below:

$$\begin{aligned}
\frac{u(x)}{v(x)} &= \frac{u_m x^m + \cdots + u_1 x + u_0}{v_n x^n + \cdots + v_1 x + v_0} \\
&= \left(u_m + \frac{u_{m-1}}{x} + \cdots + \frac{u_0}{x^m}\right) \frac{x^m}{v_n x^n + \cdots + v_1 x + v_0} \\
&= \left(u_m + \frac{u_{m-1}}{x} + \cdots + \frac{u_0}{x^m}\right) \left(s(x) + \frac{t(x)}{v_n x^n + \cdots + v_1 x + v_0}\right), \qquad \deg(t) < n.
\end{aligned}$$

So $s(x)$ and $t(x)$ are the quotient and remainder of x^m divided by $v(x)$, respectively. Now, $q(x) = \lfloor \frac{u(x)}{v(x)} \rfloor$ is completely determined by the product of $u_m + \frac{u_{m-1}}{x} + \cdots + \frac{u_0}{x^m}$ with $s(x)$. Suppose $s(x)$ is already computed, then we simply multiply $u(x)$ with $s(x)$, throw away all terms of degree less than m, and scale the resulting polynomial by x^{-m}. The result will be $q(x)$. For multiplication, we use FFT which costs time $O(m \lg m)$, or $O(n \lg n)$ if m is on the order of n.

But how do we compute $s(x) = \left\lfloor \frac{x^m}{v(x)} \right\rfloor$ efficiently? Note that we can "scale" polynomials by multiplying and dividing by powers of x easily. So we assume that $v(x)$ is of degree $n = 2^l - 1$ for some integer l. If not, we multiply both $u(x)$ and $v(x)$ by $x^{2^{\lceil \log_2^{n+1} \rceil} - 1 - n}$.

Given that the degree n of $v(x)$ is now one less than some perfect power of 2, we look at how to find the *reciprocal* $s(x)$ of $v(x)$, which is defined to be $\left\lfloor \frac{x^{2n}}{v(x)} \right\rfloor$. If $m \le 2n$, to obtain $\left\lfloor \frac{u(x)}{v(x)} \right\rfloor$, we multiply $s(x)$ with $u(x)$, discard all terms of degree less than $2n$ in the product polynomial, and finally, scale the resulting polynomial by x^{-2n}. If $m > 2n$, then we obtain

$$\begin{aligned}
\left\lfloor \frac{u(x)}{v(x)} \right\rfloor &= \left\lfloor \frac{u(x)}{x^{2n}} \left(s(x) + \frac{t(x)}{v(x)}\right) \right\rfloor, \qquad \deg(t) < n \\
&= \left\lfloor \frac{u(x)s(x)}{x^{2n}} \right\rfloor + \left\lfloor \frac{u(x)t(x)}{x^{2n}v(x)} \right\rfloor \\
&= \left\lfloor \frac{u(x)s(x)}{x^{2n}} \right\rfloor + \left\lfloor \left\lfloor \frac{u(x)t(x)}{x^{2n}} \right\rfloor \Big/ v(x) \right\rfloor.
\end{aligned}$$

To obtain the first term in the last equation above, we compute the product $u(x)s(x)$, trim off all terms of degree less than $2n$, and then scale by x^{-2n}. To obtain $\lfloor u(x)t(x)/x^{2n} \rfloor$, we compute the product $u(x)t(x)$ and carry out the same trimming and scaling steps. Then we end up with another division problem involving the new dividend $\lfloor u(x)t(x)/x^{2n} \rfloor$ and the divisor $v(x)$, where the reciprocal of $v(x)$ can be used again. The degree of the dividend has reduced by at least $n+1$ since $\deg(t) \le n-1$. The quotient of this second division will be added to the quotient obtained in the first division. And so on. As long as m is on the order of n, the procedure will terminate after a constant number of divisions.

In computing the quotient, all the multiplications can be carried out by FFT and cost $O(n \lg n)$ together. The running time of the algorithm then depends on how fast the reciprocal can be computed.

The procedure Reciprocal below takes as input a polynomial $p(x) = \sum_{i=0}^{k-1} a_i x^i$, where $a_{k-1} \neq 0$ and k is a power of 2. It computes $\lfloor x^{2k-2}/p(x) \rfloor$.

$\text{Reciprocal}\left(\sum_{i=0}^{k-1} a_i x^i\right)$

1 **if** $k = 1$

2 **then return** $1/a_0$

3 **else** $q(x) \leftarrow \text{Reciprocal}\left(\sum_{i=k/2}^{k-1} a_i x^{i-k/2}\right)$

4 $r(x) \leftarrow 2q(x)x^{(3/2)k-2} - \left(q(x)\right)^2 \left(\sum_{i=0}^{k-1} a_i x^i\right)$

5 **return** $\left\lfloor \dfrac{r(x)}{x^{k-2}} \right\rfloor$

Example 1. Let us compute $\lfloor x^{14}/p(x) \rfloor$, where

$$p(x) = x^7 - x^6 + x^5 + 2x^4 - x^3 - 3x^2 + x + 4.$$

Here $k = 8$. In line 3 of the procedure Reciprocal, a recursive call is made to compute the reciprocal of $x^3 - x^2 + x + 2$. You may verify that the recursive call returns

$$\begin{aligned} q(x) &= \left\lfloor \frac{x^6}{x^3 - x^2 + x + 2} \right\rfloor \\ &= x^3 + x^2 - 3. \end{aligned}$$

Line 4 yields

$$\begin{aligned} r(x) &= 2q(x)x^{10} - (q(x))^2 p(x) \\ &= x^{13} + x^{12} - 3x^{10} - 4x^9 + 3x^8 + 15x^7 + 12x^6 - 42x^5 - 34x^4 + 39x^3 + 51x^2 - 9x - 36. \end{aligned}$$

Then at line 5, the result is

$$s(x) = x^7 + x^6 - 3x^4 - 4x^3 + 3x^2 + 15x + 12.$$

You may verify that $s(x)p(x)$ is x^{14} plus a polynomial of degree 6.

Theorem 1 *The procedure* Reciprocal *correctly computes the reciprocal of a polynomial.*

Proof By induction on k, for k a power of 2. Namely, we prove that if $s(x) = \text{Reciprocal}(p(x))$, and $\deg(p(x)) = k - 1$, then $s(x)p(x) = x^{2k-2} + t(x)$, where $\deg(t(x)) < k - 1$. The base case $k = 1$ is trivial, since $p(x) = a_0$, $s(x) = 1/a_0$, and $t(x)$ need not exist.

For the inductive step, let $p(x) = p_1(x)x^{k/2} + p_2(x)$, where $\deg(p_1) = \frac{k}{2} - 1$ and $\deg(p_2) \le \frac{k}{2} - 1$. By the inductive hypothesis, if $s_1(x) = \textsc{Reciprocal}(p_1(x))$, then

$$s_1 p_1 = x^{k-2} + t_1(x), \tag{2}$$

where $\deg(t_1) < \frac{k}{2} - 1$. Line 4 of the procedure computes

$$r(x) = 2s_1 x^{(3/2)k-2} - s_1^2 \left(p_1 x^{k/2} + p_2 \right). \tag{3}$$

In order for the output $\lfloor r(x)/x^{k-2} \rfloor$ to be the reciprocal of $p(x)$, $r(x)p(x)/x^{k-2}$ must be x^{2k-2} plus some terms of degree less than x^{k-1}. So it suffices to show that $r(x)p(x)$ is x^{3k-4} plus terms of degree less than $2k-3$.

By (3) and the fact that $p = p_1 x^{k/2} + p_2$, we have

$$\begin{aligned}
r \cdot p &= 2s_1p_1x^{2k-2} + 2s_1p_2x^{(3/2)k-2} - \left(s_1p_1x^{k/2} + s_1p_2\right)^2 \\
&= 2\left(x^{k-2} + t_1\right)x^{2k-2} + 2s_1p_2x^{(3/2)k-2} - \left((x^{k-2} + t_1)x^{k/2} + s_1p_2\right)^2, \qquad \text{substitute (2) in} \\
&= 2x^{3k-4} + 2t_1x^{2k-2} + 2s_1p_2x^{(3/2)k-2} - x^{3k-4} - 2x^{(3/2)k-2}\left(t_1x^{k/2} + s_1p_2\right) - \left(t_1x^{k/2} + s_1p_2\right)^2 \\
&= x^{3k-4} - \left(t_1x^{k/2} + s_1p_2\right)^2.
\end{aligned}$$

Since $\deg(t_1) \le \frac{k}{2} - 2$, $\deg(s_1) = \frac{k}{2} - 1$, and $\deg(p_2) \le \frac{k}{2} - 1$, the term $\left(t_1x^{k/2} + s_1p_2\right)^2$ is of degree at most $2k-4$. $\square$

Let $T(k)$ be the running time the procedure Reciprocal on $\sum_{i=0}^{k-1} a_i x^i$. Then line 3 takes time $T(k/2)$. Line 4 can be executed in time $O(k \lg k)$ using FFT. So we set up the recurrence

$$T(k) = T\left(\frac{k}{2}\right) + O(k \lg k),$$

which has the solution $O(k \lg k)$.

Based on all the above, we have arrived at the following conclusion.

Theorem 2 *Let $u(x) = u_m x^m + \cdots u_1 x + u_0$ and $v(x) = v_n x^n + \cdots v_1 x + v_0$ be two polynomials of degrees m and n, respectively, such that $m \ge n$ and $m = \Theta(n)$. Then the quotient $q(x) = \lfloor u(x)/v(x) \rfloor$ and the remainder $r(x) = u(x) - q(x)v(x)$ can be computed in time $O(n \lg n)$.*

2 The Euclidean Algorithm

Let a_0 and a_1 be two positive integers. The *greatest common divisor* of a_0 and a_1, often denoted by $\gcd(a_0, a_1)$, divides both a_0 and a_1, and is divided by every divisor of both a_0 and a_1. Euclid's algorithm obtains $\gcd(a_0, a_1)$ by repeatedly computing $a_{i+1} = a_{i-1} - q_i a_i$, for $1 \le i < k$, where $q_i = \lfloor a_{i-1}/a_i \rfloor$.

Example 2. Let $a_0 = 501$ and $a_1 = 111$. Then Euclid's algorithm generates the following:

$$\begin{aligned}
501 &= 4 \cdot 111 + 57, \\
111 &= 1 \cdot 57 + 54, \\
57 &= 1 \cdot 54 + 3, \\
54 &= 18 \cdot 3.
\end{aligned}$$

Since the last division results in a remainder of zero, $\gcd(501, 111) = 3$. Meanwhile, we can trace back the computation, starting from the second to last division:

$$\begin{aligned} 3 &= 57 - 54 \\ &= 57 - (111 - 57) \\ &= 2 \cdot 57 - 111 \\ &= 2 \cdot (501 - 4 \cdot 111) - 111 \\ &= 2 \cdot 501 - 9 \cdot 111. \end{aligned}$$

In this way we find integers $x = 2$ and $y = -9$ such that

$$a_0 x + a_1 y = \gcd(a_0, a_1).$$

Euclid's algorithm can be extended to find not only the greatest common divisor of a_0 and a_1, but also integers x and y such that $a_0 x + a_1 y = \gcd(a_0, a_1)$. The algorithm is as follows.

```
EXTENDED-EUCLID(a_0, a_1)
1    x_0 ← 1
2    y_0 ← 0
3    x_1 ← 0
4    y_1 ← 1
5    i ← 1
6    while a_i does not divide a_{i-1}
7        q ← ⌊a_{i-1}/a_i⌋
8        a_{i+1} ← a_{i-1} - q a_i
9        x_{i+1} ← x_{i-1} - q x_i
10       y_{i+1} ← y_{i-1} - q y_i
11       i ← i + 1
```

EXAMPLE 3. For the previous example, we obtain the following values for the a_i's, x_i's, and y_i's.

i	a_i	x_i	y_i
0	501	1	0
1	111	0	1
2	57	1	−4
3	54	−1	5
4	3	2	−9

Let us use induction to show that in the procedure EXTENDED-EUCLID

$$a_0 x_i + a_1 y_i = a_i.$$

Apparently, the equation holds for $i = 0$ and $i = 1$ by lines 1–4 of the procedure. Assume that it holds for $i - 1$ and i. Then $x_{i+1} = x_{i-1} - q x_i$ by line 9 and $y_{i+1} = y_{i-1} - q y_i$ by line 10. Thus

$$a_0 x_{i+1} + a_1 y_{i+1} = a_0 x_{i-1} + a_1 y_{i-1} - q(a_0 x_i + a_1 y_i).$$

By the induction hypothesis and the above equation, we have

$$\begin{aligned} a_0 x_{i+1} + a_1 y_{i+1} &= a_{i-1} - q a_i \\ &= a_{i+1}, \qquad \text{by line 8.} \end{aligned}$$

Next, we introduce some notation that will be useful in the development of the greatest common divisor algorithm for polynomials. Let a_0 and a_1 be integers with remainder sequence $a_0, a_1, \ldots, a_k$. For $1 \le i \le k$ let $q_i = \lfloor a_{i-1}/a_i \rfloor$. We define, for $0 \le i \le j \le k$, the matrix

$$R_{ij}^{(a_0,a_1)} = R_{ij} = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{if } i = j; \\ \begin{pmatrix} 0 & 1 \\ 1 & -q_j \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & -q_{j-1} \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{pmatrix}, & \text{if } i < j. \end{cases}$$

Example 4. Let $a_0 = 501$ and $a_1 = 111$ with remainder sequences 501, 111, 57, 54, 3 and quotients q_i, for $1 \le i \le 4$, given by 4, 1, 1, 18. Then

$$\begin{aligned} R_{03} &= \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & -4 \end{pmatrix} \\ &= \begin{pmatrix} -1 & 5 \\ 2 & -9 \end{pmatrix}. \end{aligned}$$

For $i < j < k$ we have

$$\begin{aligned} \begin{pmatrix} a_j \\ a_{j+1} \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 1 & -q_j \end{pmatrix} \cdot \begin{pmatrix} a_{j-1} \\ a_j \end{pmatrix} \\ &\vdots \\ &= \begin{pmatrix} 0 & 1 \\ 1 & -q_j \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{pmatrix} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix} \\ &= R_{ij} \begin{pmatrix} a_i \\ a_{i+1} \end{pmatrix}. \end{aligned}$$

In particular,

$$R_{0j} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} a_j \\ a_{j+1} \end{pmatrix}.$$

Namely, we can use R_{0j} to directly obtain the jth and $(j+1)$-th remainders in the remainder sequence of (a_0, a_1).

Finally, we use induction to show that

$$R_{0j} = \begin{pmatrix} x_j & y_j \\ x_{j+1} & y_{j+1} \end{pmatrix}, \qquad \text{for } 0 \le j \le k.$$

The equation apparently holds when $j = 0$. Suppose it holds for some j. Then

$$\begin{aligned} R_{0,j+1} &= \begin{pmatrix} 0 & 1 \\ 1 & -q_{j+1} \end{pmatrix} R_{0j} \\ &= \begin{pmatrix} 0 & 1 \\ 1 & -q_{j+1} \end{pmatrix} \begin{pmatrix} x_j & y_j \\ x_{j+1} & y_{j+1} \end{pmatrix} \\ &= \begin{pmatrix} x_{j+1} & y_{j+1} \\ x_{j+2} & y_{j+2} \end{pmatrix}, \qquad \text{by lines 9 and 10 in Extended-Euclid.} \end{aligned}$$

3 The Procedure HGCD

Let $a_0(x)$ and $a_1(x)$ be two polynomials whose greatest common divisor we wish to compute. Assume $\deg(a_1(x)) < \deg(a_0(x))$. If their degrees are the same, replace them by a_0 and a_0 modulo a_1, or simply, $a_0 \bmod a_1$.

For polynomials over a field the greatest common divisor is unique only up to multiplication by a constant. That is, if $g(x)$ divides $a_0(x)$ and $a_1(x)$ and any other divisor of these two polynomials also divides $g(x)$, then $cg(x)$ also has this property for any constant $c \neq 0$. We shall be satisfied with finding any one greatest common divisor.[1]

The GCD algorithm will employ a divide-and-conquer strategy. We will first design an algorithm that obtains the last term in the remainder sequence whose degree is more than $\deg(a_0)/2$. Let $a_{l(i)}$ be the remainder in the sequence whose degree is greater than i but whose following remainder $a_{l(i)+1}$ has degree at most i. Since $\deg(a_i) \leq \deg(a_{i-1}) - 1$ for all $i \geq 1$, it follows that if a_0 is of degree n, then $l(i) \leq n - i - 1$.

The quotient of two polynomials of degree d_1 and d_2, with $d_1 > d_2$, has degree $d_1 - d_2$. It depends only on the leading $\min\{d_1 - d_2 + 1, d_2\}$ terms of the divisor and the leading $d_1 - d_2 + 1$ terms of the dividend. This is because the total number of shifts in carrying out the division is $d_1 - d_2$. Only the leading $d_1 - d_2 + 1$ terms of the divisor will have its multiples subtracted from the leading $d_1 - d_2 + 1$ terms of the dividend to determine the quotient.

Using the above principle, we now introduce a recursive procedure HGCD (half GCD) which takes a_0 and a_1, with $n = \deg(a_0) > \deg(a_1)$, and produces the matrix R_{0j}, where $j = l(n/2)$. Afterward, we can easily obtain $a_j = R_{0j}a_0$ as the last term in the remainder sequence whose degree exceeds $\deg(a_0)/2$.

HGCD(a_0, a_1)

if $\deg(a_1) \leq \deg(a_0)/2$

  then return $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

  else $m \leftarrow \lfloor \deg(a_0)/2 \rfloor$

    let $a_0 = b_0 x^m + c_0$, where $\deg(c_0) < m$;

    let $a_1 = b_1 x^m + c_1$, where $\deg(c_1) < m$.

    $R \leftarrow$ HGCD(b_0, b_1)

    $\begin{pmatrix} d \\ e \end{pmatrix} \leftarrow R \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$

    $f \leftarrow d \bmod e$

    let $e = g_0 x^{\lfloor m/2 \rfloor} + h_0$, where $\deg(h_0) < \lfloor m/2 \rfloor$;

    let $f = g_1 x^{\lfloor m/2 \rfloor} + h_1$, where $\deg(h_1) < \lfloor m/2 \rfloor$.

    $S \leftarrow$ HGCD(g_0, g_1)

    $q \leftarrow \lfloor d/e \rfloor$

    return $S \cdot \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} \cdot R$

[1] To insure uniqueness we could insist that the greatest common divisor be *monic*, that is, its leading term has coefficient 1.

In lines 4–5, b_0 and b_1 are the leading terms of a_0 and a_1, respectively. We have $\deg(b_0) = \lceil \deg(a_0)/2 \rceil$ and $\deg(b_0) - \deg(b_1) = \deg(a_0) - \deg(a_1)$. In lines 7–8, d, e, and f are successive terms in the remainder sequence generated from a_0 and a_1. As we will see, d is the last term of degree greater than $\lceil 3m/2 \rceil$ in the remainder sequence of a_0 and a_1; so e and f have degrees at most $\lceil 3m/2 \rceil$, that is, $\frac{3}{4}\deg(a_0)$. Also g_0 and g_1 are each of degree at most $m+1$.

Example 5. Let us first illustrate the execution of the procedure HGCD on the following polynomials:

$$\begin{aligned} p_1(x) &= x^5 + x^4 + x^3 + x^2 + x + 1, \\ p_2(x) &= x^4 - 2x^3 + 3x^2 - x - 7. \end{aligned}$$

Suppose we attempt to compute $\text{HGCD}(p_1, p_2)$; hence $a_1 = p_1$ and $a_2 = p_2$. At lines 3–5, we have $m = 2$ and

$$\begin{aligned} b_0 &= x^3 + x^2 + x + 1, \\ c_0 &= x + 1, \\ b_1 &= x^2 - 2x + 3, \\ c_1 &= -x - 7. \end{aligned}$$

At line 6, $\text{HGCD}(b_0, b_1)$ is called and returns the value

$$R = \begin{pmatrix} 0 & 1 \\ 1 & -(x+3) \end{pmatrix}$$

as we may check. Next, at lines 7–8, we compute

$$\begin{aligned} d &= x^4 - 2x^3 + 3x^2 - x - 7, \\ e &= 4x^3 - 7x^2 + 11x + 22, \\ f &= -\frac{3}{16}x^2 - \frac{93}{16}x - \frac{45}{8}. \end{aligned}$$

Since $\lfloor m/2 \rfloor = 1$, the execution of lines 9–10 yields

$$\begin{aligned} g_0 &= 4x^2 - 7x + 11, \\ h_0 &= 22, \\ g_1 &= -\frac{3}{16}x - \frac{93}{16}, \\ h_1 &= -\frac{45}{8}. \end{aligned}$$

Thus at line 11, the recursive call $\text{HGCD}(g_0, g_1)$ sets

$$S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

At line 12, the quotient $q(x)$ is found to be $\frac{1}{4}x - \frac{1}{16}$. So at line 13, we have the result

$$\begin{aligned} T &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -(\frac{1}{4}x - \frac{1}{16}) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -(x+3) \end{pmatrix} \\ &= \begin{pmatrix} 1 & -(x+3) \\ -(\frac{1}{4}x - \frac{1}{16}) & \frac{1}{4}x^2 + \frac{11}{16}x + \frac{13}{16} \end{pmatrix}. \end{aligned}$$

Note that

$$T\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} e \\ f \end{pmatrix},$$

which is correct since in the remainder sequence for p_1 and p_2, e is the last polynomial whose degree exceeds half that of p_1.

Let us consider the matrix R computed at line 6 of HGCD. Presumably Rb_0 is the last polynomial of degree greater than $\lceil m/2 \rceil$ in the remainder sequence for b_0 and b_1; that is, $R = R^{(b_0,b_1)}_{0,l(\lceil m/2 \rceil)}$. Yet, on line 7, we use R as if it were the matrix $R^{(a_0,a_1)}_{0,l(\lceil 3m/2 \rceil)}$ to obtain d and e, where d is the last term of degree greater than $\lceil 3m/2 \rceil$ in the remainder sequence of a_0 and a_1. We must show that

$$R = R^{(b_0,b_1)}_{0,l(\lceil m/2 \rceil)} = R^{(a_0,a_1)}_{0,l(\lceil 3m/2 \rceil)}.$$

Similarly, we must show that S, computed on line 11, plays the role assigned to it on line 13. That is,

$$S = R^{(g_0,g_1)}_{0,l(\lceil m/2 \rceil)} = R^{(e,f)}_{0,l(m)}.$$

Lemma 3 *Consider the following two polynomials:*

$$\begin{aligned} f(x) &= f_1(x)x^k + f_2(x), \\ g(x) &= g_1(x)x^k + g_2(x), \end{aligned}$$

where $\deg(f) \geq \deg(g)$, $\deg(f_2) < k$, *and* $\deg(g_2) < k$. *Let*

$$\begin{aligned} f(x) &= q(x)g(x) + r(x), \\ f_1(x) &= q_1(x)g_1(x) + r_1(x), \end{aligned}$$

where $\deg(r) < \deg(g)$ *and* $\deg(r_1) < \deg(g_1)$. *If* $k \leq 2\deg(g) - \deg(f)$, *namely,* $\deg(g_1) \geq \frac{1}{2}\deg(f_1)$, *then*

(a) $q(x) = q_1(x)$;

(b) $r(x)$ *and* $r_1(x)x^k$ *agree in all terms of degree* $k + \deg(f) - \deg(g)$ *or higher.*

Proof Consider dividing $f(x)$ by $g(x)$ using the ordinary division algorithm which divides the first term of $f(x)$ by the first term of $g(x)$ to get the first term of the quotient. The first term of the quotient is multiplied by $g(x)$ and subtracted from $f(x)$ and so on. The first $\deg(g) - k + 1$ terms of the quotient produced only involve the leading $\deg(g) - k + 1$ terms of $g(x)$, that is, terms of degree k or higher; thus they do not depend on $g_2(x)$. Meanwhile, the quotient has degree $\deg(f) - \deg(g)$ and thus $\deg(f) - \deg(g) + 1$ terms. Therefore if $\deg(f) - \deg(g) + 1 \leq \deg(g) - k + 1$, the quotient does not depend on $g_2(x)$. But this follows from that $k \leq 2\deg(g) - \deg(f)$. Similarly, the quotient involves only the leading $\deg(f) - \deg(g) + 1$ terms of $f(x)$. So if $\deg(f) - \deg(g) + 1 \leq \deg(f) - k + 1$, the quotient does not depend on $f_2(x)$ since $\deg(f_2) < k$. But the condition $\deg(f) - \deg(g) + 1 \leq \deg(f) - k + 1$ follows from that $k \leq 2\deg(g) - \deg(f)$ and $\deg(f) > \deg(g)$. Therefore $q(x)$ does not depend on $f_1(x)$ or $g_1(x)$ and part (a) follows.

To prove part (b), observe that the division requires $\deg(f) - \deg(g)$ shifts of $g(x)$ (that is, successive subtractions of products of $g(x)$ with terms $x^{\deg(f)-\deg(g)}, \ldots, x, 1$ scaled by constants).

So $g_2(x)$ must be shifted the same number of times. Since it has at most k terms, only $\deg(f) - \deg(g) + k$ of the remainder resulting from the division of $f(x)$ by $g(x)$ are affected by $g_2(x)$. In other words, the remainder terms of degree $\deg(f) - \deg(g) + k$ or higher do not depend on $g_2(x)$. Similarly, terms of the remainder of degree k or greater do not depend on $f_2(x)$. But $\deg(f) - \deg(g) + k > k$. Thus $r(x)$ and $r_1(x)x^k$ agree in all terms of degree $\deg(f) - \deg(g) + k$ or higher. □

Lemma 4 *Let $f(x) = f_1(x)x^k + f_2(x)$ and $g(x) = g_1(x)x^k + g_2(x)$, where deg$(g) < \deg(f) = n$, $\deg(f_2) < k$, and $\deg(g_2) < k$. Then the quotients of the remainder sequences for (f, g) and (f_1, g_1) agree at least until the latter sequence reaches a remainder of degree no more than $\deg(f_1)/2$. In other words, we have*

$$R^{(f,g)}_{0,l(\lceil (n+k)/2\rceil)} = R^{(f_1,g_1)}_{0,l(\lceil (n-k)/2\rceil)}.$$

Proof Lemma 3 assumes that the quotients agree, and in the remainder sequences for (f, g) and (f_1, g_1) a sufficient number of higher order terms agree. Use the fact that f_1 is of degree $n - k$. □

The next theorem establishes that the procedure HGCD generates all terms in the remainder sequence that have degree greater than $\frac{n}{2}$.

Theorem 5 *Let $a_0(x)$ and $a_1(x)$ be polynomials with* $\deg(a_0) = n$ *and* $\deg(a_1) < n$. *Then* $\text{HGCD}(a_0, a_1) = R_{0,l(n/2)}$.

Proof We use induction on n. By Lemma 4, R computed on line 6 in the procedure HGCD is

$$R^{(b_0,b_1)}_{0,l(\lceil m/2\rceil)} = R^{(a_0,a_1)}_{0,l(\lceil 3m/2\rceil)}.$$

Namely, $R\binom{a_0}{a_1}$ produces the last term in the remainder sequence that has degree greater than $\lceil 3m/2\rceil$. Note that g_0 and g_1 on lines 9–10 have degrees at most $2\lceil m/2\rceil$. Lemma 4 also guarantees that the S computed on line 11 is

$$R^{(g_0,g_1)}_{0,l(\lceil m/2\rceil)} = R^{(a_0,a_1)}_{l(\lceil 3m/2\rceil)+1,\, l(m)}.$$

And q computed on line 12 yields the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} = R^{(a_0,a_1)}_{l(\lceil 3m/2\rceil),\, l(\lceil 3m/2\rceil)+1}.$$

□

Roughly speaking, to compute $R^{(a_0,a_1)}_{0,n/2}$, the recursive calls to HGCD calculate $R^{(a_0,a_1)}_{0,3n/4}$, $R^{(a_0,a_1)}_{3n/4,5n/8}$, $R_{5n/8,9n/16}$, …, in the order. The lower indices of these R matrices given here are not exact as they are indeed not consecutive. Every two adjacent matrices in the sequence is joined together by the matrix $\begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix}$ on line 13.

Now let us analyze the running time of the procedure HGCD. Let $T(n)$ be the time for HGCD on inputs of degree at most n. The recursive calls on lines 6 and 11 each takes time at most $T(n/2)$.

The most expensive of the other operations are the multiplications on line 7 and the divisions on lines 8 and 12, which can be performed in time $O(n \lg n)$ using FFT. Thus we have the recurrence

$$T(n) \leq 2T\binom{n}{2} + O(n \lg n).$$

The solution is $T(n) = O(n \lg^2 n)$.

4 A Fast Algorithm for Polynomial GCD's

The algorithm for greatest common divisors uses the procedure HGCD to calculate $R_{0,n/2}$, then $R_{0,3n/4}$, then $R_{0,7n/8}$, and so on, where n is the degree of the input.

```
GCD(a0, a1)
1  if a1 divides a0
2     then return a1
3     else R ← HGCD(a0, a1)
4           (b0, b1)^T ← R (a0, a1)^T
5           if b1 divides b0
6              then return b1
7              else c ← b0 mod b1
8                   return GCD(b1, c)
```

EXAMPLE 6. Let us continue Example 5. There $p_1(x) = x^5+x^4+x^3+x^2+1$ and $p_2(x) = x^4-2x^3+3x^2-x-7$. We already found

$$\mathrm{HGCD}(p_1, p_2) = \begin{pmatrix} 1 & -(x+3) \\ -(\frac{1}{4}x - \frac{1}{16}) & \frac{1}{4}x^2 + \frac{11}{16}x + \frac{13}{16} \end{pmatrix}.$$

Thus we compute $b_0 = 4x^3 - 7x^2 + 11x + 22$ and $b_1 = -\frac{3}{16}x^2 - \frac{93}{16}x - \frac{45}{8}$ at line 4. We find that b_1 does not divide b_0. At line 7, we find

$$b_0 \bmod b_1 = 3952x + 3952.$$

Since the latter divides $-\frac{3}{16}x^2 - \frac{93}{16}x - \frac{45}{8}$, the call to GCD at line 8 terminates at line 2 and produces $3952x + 3952$ as an answer. Of course, $x + 1$ is also a greatest common divisor of p_1 and p_2.

Let $T(n)$ be the running time of the procedure GCD on input polynomials of degree n. Since $\deg(b_1) \leq \deg(a_0)/2$, so the recursive call of GCD on line 8 takes time $T(n/2)$. The divisions and multiplications on lines 1, 4, 5, 6 together require time $O(n \lg n)$. The call to HGCD takes time $O(n \lg^2 n)$. Therefore we arrive at the following recurrence

$$T(n) \leq T\left(\frac{n}{2}\right) + O(n \lg n) + O(n \lg^2 n).$$

Thus the greatest common divisor of two polynomials of degree at most n can be computed in $O(n \lg^2 n)$ time.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] D. E. Knuth. *Seminumerical Algorithms*, vol. 2 of *The Art of Computer Programming*, 3rd edition. Addison-Wesley, 1998.