Foundations of Computer Algebra: Assignments 2 & 3

Marc Moreno Maza
University of Western Ontario
CS424 / CS556 – Winter 2008

Posted: Tuesday, March 18, 2008
Due: Tuesday, April 8, 2008

1 Guidelines

Date of postage. Tuesday, March 18, 2008.

Normal date for submission. Tuesday, April 8, 2008.

Due format. The assignment can be returned to the instructor by email in a form of a postscript, PDF or text file. Alternatively, a printed version can be delivered to his mail box in the Computer Science Department or given to him in class.

Collaboration. You are expected to do this assignment on your own without assistance from anyone else in the class. However, you can use the literature, and in particular the book *Modern Computer Algebra* and the references therein.

Marking. This assignment will be marked out of 40. Exercises 1,4,5 are worth 10 points each whereas Exercises 2,3 are worth 5 points each.

2 Exercise 1

2.1 Statement

Let $p > 2$ be a prime integer. Let f and g in $\mathbb{Z}/p\mathbb{Z}[x]$ be two polynomials with respective degrees $m \geq n > 0$. We aim at computing the product $f\,g$ by a fast algorithm. We consider here the case where $m+n+1$ does not divide $p-1$. Hence, there are no $(n+m+1)$-th primitive roots of unity in $\mathbb{Z}/p\mathbb{Z}$ and the FFT-based algorithm studied in class does not apply in this context. However, we assume that we can find r primes numbers $2 < p_1 < \cdots < p_r$ such that:

- the product m of these primes exceeds $C = (n+1)(\frac{p-1}{2})^2$,
- every prime $p_1, \ldots, p_r$ fits in a machine word,
- for all $i = 1, \ldots, r$, the integer $m+n+1$ is a power of 2 and divides $p_i - 1$, such that, regarding f, g as polynomials in $\mathbb{Z}/p_i\mathbb{Z}[x]$, one can compute the product fg in $\mathbb{Z}/p_i\mathbb{Z}[x]$ using the FFT-based algorithm studied in class.

Based on these hypotheses, we are to design a modular algorithm for computing fg in $\mathbb{Z}/p\mathbb{Z}[x]$.

2.2 Questions

(1) Show that C is an upper bound for the coefficients of $f\,g$ computed in $\mathbb{Z}[x]$ (when regarding f and g as polynomials of $\mathbb{Z}[x]$).

(2) Describe a modular algorithm for computing $f\,g$ in $\mathbb{Z}[x]$ **and** give an upper for the number of machine word operations required by this algorithm.

(3) Deduce an algorithm for computing $f\,g$ in $\mathbb{Z}/p\mathbb{Z}[x]$ **and** give an upper for the number of machine word operations required by this algorithm.

2.3 Hints

For Question (3), you will need to take into account the cost of reducing modulo p the product $f\,g$ computed in $\mathbb{Z}[x]$, which means taking into account the cost of reducing each of these coefficients modulo p.

3 Exercise 2

3.1 Statement

Let $F = \mathbb{Z}/17\mathbb{Z}$ and $f = 5x^3 + 3x^2 - 4x + 3, g = 2x^3 - 5x^2 + 7x - 2$ in $F[x]$.

3.2 Questions

(1) Show that $\omega = 2$ is a primitive 8-th root of unity in F, and compute the inverse ω^{-1} of ω in F.

(2) Trace the algorithm FFT (Algorithm 2 in Section 1.5 in lecture notes) and its subroutine DFT (Algorithm 1 in Section 1.4 in lecture notes) to compute the product $h = fg \in F[x]$ with Maple, where ω is given as above.

4 Exercise 3

4.1 Statement

Let $\mathbb{R}$ be a commutative ring and let $n > 0$ be a power of 2. Let ω be a primitive $2n$-th root of unity. We consider the following polynomial $m = x^n - x - 1$ in $\mathbb{R}[x]$. Let $f = \Sigma_{k=0}^{k=n-1} f_k x^k$ and $g = \Sigma_{k=0}^{k=n-1} g_k x^k$ be two polynomials of degree less than n. The goal of this exercise is to show that one can compute $fg \mod m$ in $6n \log n + O(n)$ operations in $\mathbb{R}$. Let q and r be the quotient and the remainder in the division of fg by m. We define the polynomial $c = -qx + r$.

4.2 Questions

(1) Show that c can be computed in $\frac{9}{2} n \log n + O(n)$ operations in $\mathbb{R}$.

(2) Reusing the results of the intermediate calculations of (1), show that no more than $\frac{3}{2} n \log n + O(n)$ operations in $\mathbb{R}$ are needed for computing q.

(3) Deduce that r can be computed in $6n \log n + O(n)$ operations in $\mathbb{R}$.

(4) Give a sharp estimate for the *naive approach*, that is for:

1. computing the product fg in $\mathbb{R}[x]$ and then
2. computing the division of fg by m.

5 Exercise 4

5.1 Statement

Let $\mathbb{R}$ be a commutative ring. Let $d \longmapsto M(d)$ be a *multiplication time*. Hence, for any two polynomials $f, g \in \mathbb{R}[x]$ of degree less than d one can compute the product fg in at most $M(d)$ operations in $\mathbb{R}$. Moreover, for all $d' \geq d \geq 0$ we have $M(d') + M(d) \leq M(d' + d)$.

Let $d' \geq d > 0$ be two integers. Let $f \in \mathbb{R}[x]$ be of degree d' and let $m_1, \ldots, m_r \in \mathbb{R}[x]$ be monic polynomials such that the sum of their degrees $\deg m_1 + \cdots + \deg m_r$ is d.

The goal of this exercise is to show that all remainders $f \mod m_1, \ldots, f \mod m_r$ can be computed in $O(M(d) \log d + M(d'))$ operations in $\mathbb{R}$.

The following result (proved in class) will be essential: division-with-remainder of a polynomial $a \in \mathbb{R}[x]$ of degree $n + m$ by a monic polynomial $b \in \mathbb{R}[x]$ of degree n (where n, m are non-negative integers) can be done in at most $D(m, n)$ ring operations in $\mathbb{R}$ where we have

$$D(m,n) = \begin{cases} 4M(m) + M(n) + O(n) & \text{if} \quad m \leq n, \\ 5M(m) + O(m) & \text{if} \quad m \geq n. \end{cases}$$

5.2 Questions

(1) Consider the particular case where $d' = 4\,d$, $r = 2$ and $M(d) = 2d^2$. Compare the costs of the following two approaches:

1. compute first the remainder $r_{12} = f \mod m_1 m_2$ and then, the remainders $r_1 = r_{12} \mod m_1$ and $r_2 = r_{12} \mod m_2$,
2. compute the remainders $f \mod m_1$ and $f \mod m_2$ directly.

(2) Using a divide-and-conquer strategy, show that all remainders $f \mod m_1, \ldots, f \mod m_r$ can be computed in $O(M(d) \log d)$ operations in $\mathbb{R}$.

6 Exercise 5

6.1 Statement

Let $\mathbb{K}$ be a field and let $m \in \mathbb{K}[x]$ be a polynomial of degree $d \geq 2$. We aim at computing in $\mathbb{K}[x]/\langle m \rangle$ as fast as possible. Let $\ell \in \mathbb{K}[x]$ be another polynomial with degree strictly less than d and such that ℓ and m are relatively prime. Let $u, v \in \mathbb{K}[x]$ such that

$$u\,\ell + v\,m = 1, \quad \deg(u) < \deg(m), \quad \text{and} \quad \deg(v) < \deg(\ell).$$

We consider the application Φ from $\mathbb{K}[x]/\langle m \rangle$ to itself mapping $\overline{a}$ to $\overline{\ell a}$. We shall see that $\overline{u}\Phi(\overline{a})\Phi(\overline{b})$ can be computed in $3M(d) + O(d)$ operations in$\mathbb{K}$ for a suitable choice of ℓ. For $\overline{a}, \overline{b} \in \mathbb{K}[x]/\langle m \rangle$ define $\alpha = \Phi(\overline{a})$ and $\beta = \Phi(\overline{b})$. Then,

- define $\gamma = \overline{u}\Phi(\overline{a})\Phi(\overline{b})$.
- let (q_1, f_1) be the quotient-and-remainder in the division of $\alpha\beta$ by ℓ,
- let (q_2, f_2) be the quotient-and-remainder in the division of $v f_1$ by ℓ,
- let $f_3 := f_1 - m\, f_2$

6.2 Questions

(1) Why do we care about $\overline{u}\Phi(\overline{a})\Phi(\overline{b})$?

(2) Give a running time estimate for the computation of $\overline{u}\Phi(\overline{a})\Phi(\overline{b})$

(3) Show that ℓ divides f_3 and let f_4 be the quotient.

(4) Show that $\gamma = q_1 + f_4$ holds

(5) How to choose ℓ such that γ can be computed in $3M(d) + O(d)$ operations in$\mathbb{K}$?