Foundations of Computer Algebra. CS424b: Quiz 8. UWO, February 21, 2007.

Student ID number:
Student Last Name:

Guidelines. Every printed document is allowed but computers and calculators are not. The duration of the quiz is 1 hour. The quiz consists of three exercises located on pages 3, 4, 6. All answers should be written in the *answer boxes*. No justifications for the answers are needed, unless explicitly required. You are expected to do this quiz on your own without assistance from anyone else in the class. If possible, please avoid pencils and use pens with dark ink. Thank you.

Notations. The set of non-negative integer numbers, integer numbers and rational numbers are denoted respectively $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$, as usual. For an integer $n \in \mathbb{Z}$ the set of the multiples of n (all of them: positive, null or negative) is denoted by $n\mathbb{Z}$. For an integer $n \in \mathbb{Z}$ the residue class of the binary relation $(a, b) \longmapsto a \equiv b \mod n$ is denoted by $\mathbb{Z}/n\mathbb{Z}$, as usual. For a ring $\mathbb{A}$, we denote by $\mathbb{A}[x]$ the ring of univariate polynomials in x with coefficients in $\mathbb{A}$.

Formula: the Chinese Remaindering Theorem Let m and n be two relatively prime elements of an Euclidean domain R. (You may think of $R = \mathbb{Z}$ or $R = \mathbb{Q}[x]$.) Let $s, t \in R$ be such that $s\,m + t\,n = 1$. For every $a, b \in R$ there exists $c \in R$ such that

$$(\forall x \in R) \quad \left\{ \begin{array}{l} x \equiv a \mod m \\ x \equiv b \mod n \end{array} \right. \iff x \equiv c \mod m\,n \tag{1}$$

where a convenient c is given by

$$c \ = \ a + (b-a)\,s\,m = b + (a-b)t\,n \tag{2}$$

Therefore, for every $a, b \in R$ the system of equations

$$\left\{ \begin{array}{l} x \equiv a \mod m \\ x \equiv b \mod n \end{array} \right. \tag{3}$$

has a solution.

Formula: Fermat's little Theorem If $p \in \mathbb{N}$ is a prime and $a \in \mathbb{Z}$ then we have

$$a^p \equiv a \mod p \tag{4}$$

Moreover if p does not divide a then we have $a^{p-1} \equiv 1 \mod p$.

Formula: complexity estimates for divide-and-conquer algorithms Let $T(n)$ be the running time estimate of a divide-and-conquer algorithm. Assume that $T(n)$ satisfies a relation of the form:

$$T(n) \leq a\,T(n/b) + S(n) \tag{5}$$

where $S(n)$ is the cost of the combine-part, $a \geq 3$ is the number of recursive calls and n/b is the size of a sub-problem, with b an integer greater than 1. Assume that the following holds for all n:

$$S(b\,n) \geq bS(n) \text{ and } S(n) \geq n.$$

Then we have $T(n) \in \mathcal{O}(S(n)\,n^{\log_b(a)-1})$.

Exercise 1. Solve in $\mathbb{Z}$ the following equations or systems of equations.

(1) $7x + 15y = 1$, with unknown (x, y),

(2) $$\begin{cases} x \equiv 2 \mod 3 \\ x \equiv 4 \mod 5 \end{cases}$$

(3) $$\begin{cases} x \equiv 2 \mod 3 \\ x \equiv 4 \mod 5 \\ x \equiv 8 \mod 7 \end{cases}$$

Answer 1

Exercise 2. Let $p \in \mathbb{N}$ be a prime and $a \in \mathbb{Z}$ be a non-zero integer such that p does not divide a. It follows from Fermat's little Theorem that the inverse of $a \mod p$ can be computed by

$$a^{-1} \equiv a^{p-2} \mod p \tag{6}$$

(1) Explain why this leads to an algorithm requiring $\mathcal{O}(\log_2^3(p))$ word operations.

(2) Is this better than the modular inversion via Euclide's Algorithm?

Answer 2

Answer 3

Exercise 3. Let F and G be two non-constant univariate polynomials with variable x and with coefficients in $\mathbb{Z}/p\mathbb{Z}$ for $p > 2$. (In fact, what follows would work with any coefficient ring $\mathbb{A}$ containing at least 4 elements and where 2 is invertible.) Let $n = 3^k$, with $k \in \mathbb{N}$, be the smallest power of 3 such that F and G have degree less than n, but at least one of them has degree greater than or equal to $n/3$. (**For simplicity**, you can assume that F and G have the same degree $n-1$ and that n is a power of 3.) Hence we can write

$$F = f_2 t^2 + f_1 t + f_0 \ \text{ and } \ G = g_2 t^2 + g_1 t + g_0 \tag{7}$$

where $f_2, f_1, f_0, g_2, g_1, g_0$ are polynomials of degree less than $n/3$. Let H be the product of F and G. Our goal is to design a fast algorithm for computing H. The principle is similar to Karatsuba's trick. However, the construction is a bit more involved and leads to an asymptotically faster algorithm.

(1) Show that there exist polynomials h_4, h_3, h_2, h_1, h_0 of degree less than $2n/3$ such that we have

$$H = h_4 t^4 + h_3 t^3 + h_2 t^2 + h_1 t + h_0.$$

Let us regard F, G, H as polynomials in t with coefficients in $\mathbb{Z}/p\mathbb{Z}[x]$. We denote these polynomials by $F(t)$, $G(t)$, $H(t)$; they have respective degrees 2, 2 and 4. Observe that we have $F(t) = G(t)H(t)$. We evaluate $F(t)$ and $G(t)$ at $t = 0$, $t = 1$, $t = -1$ and $t = 2$. Moreover, we define $H(\infty) = x_2 y_2$. Hence we have $H(\infty) = w_4$.

(2) Show that computing $H(0)$, $H(1)$, $H(-1)$, $H(2)$ and $H(\infty)$ requires:

- 5 multiplications of polynomials with degree less than $n/3$,
- 12 additions or subtractions of polynomials with degree less than $n/3$,
- 4 multiplications of a polynomial with degree less than $n/3$ by a power of 2 (actually 2 or 4).

(3) Show that h_4, h_3, h_2, h_1, h_0 can be computed from $H(\infty)$, $H(2)$, $H(-1)$, $H(1)$, $H(0)$, by solving the following system of linear equations:

$$\left\{ \begin{array}{rcl} H(0) & = & h_0 \\ H(1) & = & h_4 + h_3 + h_2 + h_1 + h_0 \\ H(-1) & = & h_4 - h_3 + h_2 - h_1 + h_0 \\ H(2) & = & 16h_4 + 8h_3 + 4h_2 + 2h_1 + h_0 \\ H(\infty) & = & h_4 \end{array} \right. \tag{8}$$

(4) Show that solving this linear system can be done in $\mathcal{O}(n)$ operations in $\mathbb{Z}/p\mathbb{Z}$.

Therefore, we have obtained a method for computing the "coefficients" w_4, w_3, w_2, w_1, w_0 of $H(t)$. This provides, in fact, an algorithm for multiplying univariate polynomials in $\mathbb{Z}/p\mathbb{Z}[x]$. Let us estimate its time complexity. Let $T(n)$ be the number of operations in $\mathbb{Z}/p\mathbb{Z}[x]$ that are needed for computing H by means of this algorithm.

(5) Show that $T(n)$ satisfies a relation of the form

$$T(n) \leq 5T(n/3) + cn$$

where c is a constant.

(6) Give an asymptotic upper bound for $T(n)$ and compare with the time complexity of Karatsuba's multiplication.

Answer 4

Answer 5

Answer 6

Answer 7