Foundations of Computer Algebra. CS424b: Quiz 1. UWO, March 2, 2005.

Student ID number:
Student Last Name:

Guidelines. Every printed document is allowed but computers and calculators are not. The duration of the quiz is 40 minutes. The quiz consists of three exercises located on pages 2, 3, 4. 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 $\mathbb{A}$.

Formulas. Let m and n be two relatively prime elements of an Euclidean domain R. (You may think $R = \mathbb{Z}$ 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. We give also

$$1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{2^{2k}} < 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots + \frac{1}{2^{2k}} < 2 \tag{4}$$

Exercise 1. Let n be an even positive integer. Given two polynomials $r_0, r_1 \in \mathbb{Q}[x]$ with respective degrees n and $n/2$, we denote by $D(n)$ the number of operations (additions, subtractions, multiplications, inversions) in $\mathbb{Q}$ that are needed in order to compute the quotient q_2 and the remainder r_2 of r_0 by r_1.

1. What is the degree of q_2? Give an estimation for $D(n)$.
2. Assume now that n is a power of 2, say $n = 2^k$. We run the Euclidean Algorithm starting from r_0, r_1. Assume that the degree of each remainder r_{i+1} is half the degree of the previous remainder r_i. Is the Euclidean Algorithm faster in this special case than in the usual one? In other words, does the time complexity of the Euclidean Algorithm becomes sub-quadratic, i.e. $O(n^\alpha)$ for some $\alpha < 2$?

Answer 1

1. We have $\deg(q_2) = \deg(r_0) - \deg(r_1) = n/2$. Define $q_2 = c_{n/2}x^{n/2} + \cdots + c_1x + c_0$. The Euclidean division of r_0 by r_1 computes the quotient q_2 and the remainder r_2 as follows. Let a be the *current remainder*, initially this is r_0. For computing the coefficient c_i in q_2 one performs
   - 1 inversion (to compute c_i as the quotient of the leading coefficient of a by the leading coefficient of the divisor r_1).
   - $\deg(r_1)$ multiplications (to compute the product of c_i with each coefficient of the divisor except its leading coefficient),
   - $\deg(r_1)$ subtractions (to subtract $c_ix^i\, r_1$ from the current remainder, observing that we do not need to subtract the leading coefficient of $c_ix^i\, r_1$ from that of a, since we know that this difference is null).

   This leads to $(\deg(q_2) + 1)(2\deg(r_1) + 1) = (n/2 + 1)(n + 1)$ operations in $\mathbb{Q}$. Therefore, we have $D(n) = (n/2 + 1)(n + 1)$.
2. $D(n)$ is quadratic in n. So, the cost of this division is in the same order as the whole Euclidean Algorithm. Therefore, this special case does not lead to any speed-up. One can be curious in computing the sum $D(2^k) + D(2^{k-1}) + \cdots + D(2^1)$. Using MAPLE, invoking the SUM command, one obtains

   $$k + 2/3n^2 + 3n - 11/3.$$

Exercise 2. Let k, n be two non-negative integers with $n > 0$. Let $a, b \in \mathbb{Q}[x]$ with respective degrees $n + k$ and n. Let $D(n, k)$ the number of operations in $\mathbb{Q}$ needed in order to compute the quotient q and the remainder r of a by b The polynomial b may not be monic.

1. If $k = 0$, how many operations in $\mathbb{Q}$ are needed to compute q only? What is $D(n, 0)$?

2. If $k = 1$, show that one can compute q in at most 4 operations in $\mathbb{Q}$. What is $D(n, 1)$?

3. How many operations in $\mathbb{Q}$ for computing q only if $k = 2$? What is $D(n, 2)$?

4. Let $Q(n, k)$ be the number operations in $\mathbb{Q}$ for computing q only. Estimate $Q(n, k)$ and $D(n, k)$.

Answer 2

First, observe that the degree of q is k. Let us write $q = q_k x^k + \cdots + q_1 x + q_0$, $a = a_{n+k} x^{n+k} + \cdots + a_n x^n + \cdots + a_1 x + a_0$ and $b = b_n x^n + \cdots + b_1 x + b_0$.

1. If $k = 0$, computing $q = q_0$ which is a_n/b_n involves only one division. Hence, $Q(n, 0) = 1$. Computing both q and $r = a - q_0 b$ adds $2n$ operations (multiplications, subtractions). Hence $D(n, 0) = 2n + 1$.

2. If $k = 1$, computing q only, reduces to compute q_1 and q_0. q_1 is just a_{n+1}/b_n. For computing q_0, we just need the coefficient in x^n of

$$a - q_1 x b = (a_{n+1} x^{n+1} + a_n x^n + \cdots) - q_1 x (b_n x^n + b_{n-1} x^{n-1} + \cdots)$$

, that is $a_n - q_1 b_{n-1}$. Then, we have $q_0 = \frac{a_n - q_1 b_{n-1}}{b_n}$. Hence $Q(n, 1) = 1 + 3 = 4$. Computing both q and r implies

- computing q_1, so 1 division,
- computing the (whole) $a - q_1 x b$, costing $2n$ operations (multiplications, subtractions),
- computing q_0, so 1 division,
- computing $r = (a - q_1 x b) - q_0 b$, costing $2n$ operations.

Hence $D(n, 1) = (2n + 1)2$.

Answer 3

1. If $k = 2$, computing q only, reduces to compute q_2, q_1 and q_0. The leading coefficient q_2 is just a_{n+2}/b_n. Computing q_1 and q_0 requires the knowledge of the coefficients in x^{n+1} and x^n of

$$\begin{array}{rcl} a - q_2x^2b & = & (a_{n+2}x^{n+2} + a_{n+1}x^{n+1} + a_nx^n + \cdots) - \\ & & q_2x^2(b_nx^n + b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots) \\ & = & (a_{n+1}x^{n+1} + a_nx^n + \cdots) - \\ & & q_2x^2(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots) \\ & = & (a_{n+1} - q_2b_{n-1})x^{n+1} + (a_n - q_2b_{n-2})x^n + \cdots \end{array}$$

This amounts to 5 operations (including the computation of q_2). Now, we are back to the case $k = 1$. Hence $Q(n, 2) = 5 + Q(n, 1) = 5 + 3 + 1 = 9$. In constrast, computing both q and r implies

- computing q_2, so 1 division,
- computing the (whole) $a - q_2x^2b$, costing $2n$ operations (multiplications, subtractions),
- computing q_1, so 1 division,
- computing $(a - q_2x^2b) - q_1xb$, costing $2n$ operations,
- computing q_0, so 1 division,
- computing $((a - q_2x^2b) - q_1xb) - q_0b$, costing $2n$ operations,

Hence $D(n, 2) = (2n + 1)3$.

In the general case, one needs to understand the relation between $Q(n, k+1)$ and $Q(n, k)$. The key observation is that, if $\deg(a) = n + k + 1$, then computing $q_0, q_1, \ldots, q_k$ requires the knowledge of the coefficients of $x^{n+k}, \ldots, x^n$ in $a - q_{k+1}x^{k+1}b$, which requires $2(k + 1) + 1$ operations (including the computation of q_{k+1}). Hence, we have $Q(n, k + 1) = 2(k + 1) + 1 + Q(n, k)$. This leads to

$$Q(n, k) = \Sigma_{i=0}^{i=k} (2i + 1) = (k + 1)^2.$$

Estimating $D(n, k)$ is much easier. This is just

$$D(n, k) = (\deg(q) + 1)(2\deg(b) + 1) = (k + 1)(2n + 1).$$

Hence, if $k < n$ holds, one can save a substantial time by avoiding the computation of r.

Exercise 3. Let p, q two different prime integers. Let a be another integer, not necessarily prime and such that $a \neq 0$ and neither p nor q divides a.

1. Explain briefly why there exists two integers s, t such that $sp+tq = 1$ holds. How can we compute them?
2. Explain briefly why a has an inverse b modulo p and an inverse c modulo q. How can we compute them?
3. Explain briefly why a has an inverse d modulo pq.
4. Assume that s, t, b, c have been computed. Explain briefly at least two ways to compute d.
5. Which way do you feel to be the best one?

Answer 4

1. The integers p and q are two different primes. So, their gcd is one. Hence, the extended Euclidean algorithm computes s, t such that $sp + tq = 1$ holds.
2. Since p is a prime number and p does not divide a, the gcd of p and a is 1. Hence, the extended Euclidean algorithm computes b, v such that $ba+vp = 1$ holds. Thus b is the inverse of a modulo p. Using the extended Euclidean algorithm again, one can show the existence and compute the inverse c of a modulo q (obtaining a relation of the form $ca + wq = 1$).
3. We have $\gcd(a, pq) = 1$. Indeed, $\gcd(a, pq)$ must divide $\gcd(a, p)\gcd(a, q) = 1$.
4. The inverse d of a modulo pq can be computed by means of the extended Euclidean algorithm agin! We can also use the Chinese Remainder Theorem. Indeed, we have

   $$ba + vp = 1 \;\text{ and }\; ca + wq = 1$$

   Defining $d = b + (c - b)\, sp = b\, tq + c\, sp$ (we use the recalled formulas) we have

   $$da \equiv 1 \times 1 + 0 \equiv 1 \mod p \;\text{ and }\; da \equiv 0 + 1 \times 1 \equiv 1 \mod q$$

   Hence $da \equiv 1 \mod pq$ holds. (Indeed there exists k, k' such that $da - 1 = kp = k'q$; using Gauss Lemma p divides k'.)
5. Using the *Chinese Remainder Theorem* approach is probably faster. Indeed, computing d in this manner requires 2 additions and 2 multiplications whereas the *extended Euclidean algorithm* approach could require more arithmetic operations.