

# Assignment 1: Algorithmic Algebra with Applications

Neeraj Kayal

February 6, 2008

*NOTE: This assignment will **not** be graded. But you are encouraged to submit it and get feedback. Due Date for submission is Tuesday, 12th Feb.*

## Notation

For an object  $a$ , we will denote by  $\langle a \rangle$  the natural encoding of the object  $a$  and  $|\langle a \rangle|$  will then denote the size in bits of this representation. Thus for an integer  $a$ ,  $\langle a \rangle$  denotes the binary representation of the integer  $a$  and  $|\langle a \rangle|$  is then  $\log |a|$ . For a polynomial  $f(x)$  of degree  $d$  whose coefficients are integers of bounded by  $H$ ,  $\langle f(x) \rangle$  is the representation of  $f(x)$  as a  $(d + 1)$ -tuple of integers of magnitude at most  $H$ . Thus  $|\langle f(x) \rangle| = (d + 1) \cdot (\log H)$ . For a polynomial  $a(x)$ ,  $\deg(a(x))$ , abbreviated  $\deg(a)$ , represents the degree of the polynomial.

## Algebra Exercises

1. Show that if  $n \in \mathbb{Z}$  then for every integer  $a$  with  $\gcd(a, n) = 1$ , there exists a unique  $x$  modulo  $n$ , such that  $a \cdot x \equiv 1 \pmod{n}$ .

**Remark.** This  $x$ , referred to as  $a$ -inverse is denoted as  $a^{-1}$  or  $\frac{1}{a}$ . For two integers  $a$  and  $b$  in  $\mathbb{Z}_n$  we use  $(\frac{b}{a})$  to denote  $b \cdot a^{-1}$ .

2. Here is an age-guessing game you might play at a party. You ask a fellow party-goer to divide his age by each of the numbers 3, 4 and 5 and tell you the remainders. Show how to use this information to determine the age.
3. For an integer  $n > 1$ , let  $\phi(n)$  denote the number of positive integers less than  $n$  and coprime to  $n$ .
  - If  $m$  and  $n$  are coprime then  $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$ .
  - If  $n$  is a prime-power,  $n = p^e$ , then  $\phi(n) = (p^e - p^{e-1})$ .
  - Use the two results above to deduce a formula for  $\phi(n)$  in terms of the prime factorization of  $n$ .

4. For every pair of rings below, determine if they are isomorphic or not. If they are not isomorphic then prove this. If they are isomorphic then compute an isomorphism from the ring  $R_1$  to  $R_2$  and its inverse map.

- $R_1 = \mathbb{Z}_{15}$  and  $R_2 = \mathbb{Z}_3 \oplus \mathbb{Z}_5$ .

- $R_1 = \mathbb{Z}_{25}$  and  $R_2 = \mathbb{Z}_5 \oplus \mathbb{Z}_5$ .
- $R_1 = \mathbb{Q}[x]/\langle x^2 - 7 \rangle$  and  $R_2 = \mathbb{Q}[x]/\langle x^2 - 2 \rangle$ .
- $R_1 = \mathbb{R}[x]/\langle x^2 - 1 \rangle$  and  $R_2 = \mathbb{R}[x]/\langle x^2 + 1 \rangle$ .
- $R_1 = \mathbb{C}[x]/\langle x^2 - 1 \rangle$  and  $R_2 = \mathbb{C}[x]/\langle x^2 + 1 \rangle$ .
- $R_1 = \mathbb{Q}[x]/\langle x^3 - 1 \rangle$  and  $R_2 = \mathbb{Q}[x]/\langle x^3 + x^2 - 2x \rangle$ .
- $R_1 = \mathbb{Z}_7[x]/\langle x^3 - 1 \rangle$  and  $R_2 = \mathbb{Z}_7[x]/\langle x^3 + x^2 - 2x \rangle$ .

## Computational Complexity Exercises

1. **Robustness of the class P.** Show that if a function  $f(x)$  can be computed in polynomial time using an oracle for another polynomial-time computable function  $g(x)$  then  $f(x)$  can be computed in polynomial-time without using any oracles.
2. **Robustness of the class NC.** Show that if a function  $f(x)$  has an NC algorithm using an oracle for another NC-computable function  $g(x)$  then  $f(x)$  itself can be computed in NC without using any oracles.

**Remark.** These two exercises show that ‘the composition’ of two functions in the class P is again in the class P and the composition of two functions in the class NC is again in the class NC. This ‘closure under composition property’ makes these two classes more amenable to theoretical investigations.

3. Solve the following recurrence relations in terms of  $n$ , expressing the resulting functions asymptotically. ( $c$  represents a constant.)
  - $T(n) = 2 \cdot T(\frac{n}{2}) + cn$ .
  - $T(n) = 3 \cdot T(\frac{n}{5}) + cn$ .
  - $T(n) = 2 \cdot T(n - 1)$ .

## Algorithmic Algebra

1. **Extended Euclid Algorithm.** Let  $a, b \in \mathbb{Z}$ . Recall that if  $\gcd(a, b) = d$  then there exist integers  $x, y$  such that  $ax + by = d$ . Devise an algorithm that given  $a$  and  $b$ , with  $a > b$ , computes  $d$  and an  $x$  and a  $y$  in time  $O(|\langle a \rangle| |\langle b \rangle| + |\langle b \rangle|^3)$ .

Show that an analogous algorithm works for polynomials and assuming that all the underlying field operations are done in constant time, the time complexity for this is  $O(\deg(a) \cdot \deg(b) + \deg(b)^3)$ .

2. **Modular exponentiation.** Show that given integers  $a, t$  and  $m$ , we can compute  $a^t \pmod{m}$  in time  $O(|\langle a \rangle| \cdot |\langle m \rangle| + |\langle m \rangle|^2 |\langle t \rangle|)$ .