AM 106: Applied Algebra **Prof. Salil Vadhan**

Problem Set 8

Assigned: Fri. Nov. 16, 2018 Due: Fri. Nov. 30, 2018 (11:59pm sharp)

- You must submit your problem sets electronically on the course Canvas site. If you use LaTeX, please submit both the source (`.tex`) and the compiled file (`.pdf`). Name your files `PS8-yourlastname`.
- Aim for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details.

Problem 1. (Ideals and Factor Rings, 22pts) For each of the following rings R and subsets $I \subseteq R$, determine whether or not I is an ideal of R. If I is an ideal, do the following:

- Find a set of generators for I of minimal size, and determine whether I is principal.
- Determine the factor ring R/I by giving an appropriate homomorphism from R to a familiar ring S.
- Determine whether I is maximal, and if not, find a maximal ideal containing I.

1. $R = \mathbb{Q}$, $I = \mathbb{Z}$.
2. $R = \mathbb{Z} \times \mathbb{Z}$, $I = \{(a, b) : b \text{ even}\}$.
3. $R = \mathbb{Z}[i]$, $I = \{a + bi : b \text{ even}\}$.
4. $R = \mathbb{Z}[x]$, $I = \{p(x) : p(0) \bmod 10 = 0\}$.
5. $R = \mathbb{Q}[x]$, $I = \{p(x) : p(2) = 0 \text{ and } p(3) = 0\}$.
6. $R = \mathbb{Q}[x]$, $I = \{p(x) : p(2) = 0 \text{ or } p(3) = 0\}$.

Problem 2. (Frobenius homomorphism, Gallian 15.44+, 23pts) Let R be a commutative ring with unity and characteristic p, for a prime p.

1. Show that the map $\varphi(x) = x^p$ is a ring homomorphism from R to itself.
2. Show that if R is a finite field, then φ is an automorphism of R (i.e. an isomorphism of R with itself). (Hint: show that in this case, it suffices to prove $\ker(\varphi) = \{0\}$.)
3. Find a ring R of characteristic p such that φ is not an automorphism of R. (Hint: look at infinite R.)

Problem 3. (Polynomial Arithmetic, 10pts)

1. Let $f(x) = 5x^4 + 3x^3 + 1$ and $g(x) = 3x^2 + 2x + 1$ in $\mathbb{Z}_7[X]$. Determine the quotient and remainder upon dividing $f(x)$ by $g(x)$.

2. Show that the polynomial $2x + 1$ in $\mathbb{Z}_4[x]$ has a multiplicative inverse in $\mathbb{Z}_4[x]$.

Problem 4. (Efficiently Finding Roots, 24pts) As discussed in class, there are efficient algorithms known for factoring polynomials (in contrast to integer factorization). In this problem, you will see an efficient algorithm for the related problem of *root-finding*: given a polynomial $f(x) \in \mathbb{F}[x]$, find all $\alpha \in \mathbb{F}$ such that $f(\alpha) = 0$. This is equivalent to finding all the degree 1 factors of f. Of course, root finding is easy when $\mathbb{F}$ is small, as we can simply evaluate f at all the elements of f. When the degree of f is at most 4, there are closed-form formulas for the roots of f (like the quadratic formula for degree 2), but it is known that there are no such formulas for degree 5 and higher (see Gallian). Here you will see an efficient algorithm that works even over large finite fields (e.g. if $\mathbb{F} = \mathbb{Z}_q$ where q is an n-bit prime, the algorithm runs in time $\mathrm{poly}(n, \deg(f))$). The algorithm will be randomized, in that it will toss coins and only has polynomial running time in expectation. In what follows, let $\mathbb{F}$ be a finite field of *odd* order q, and let $f(x) \in \mathbb{F}[x]$.

1. Show that $\gcd(f(x), x^{(q-1)/2} - 1) = \prod_{\alpha \in \mathrm{QR}(\mathbb{F}^*) \cap f^{-1}(0)} (x - \alpha)$ and that $\gcd(f(x), x^{(q-1)/2} + 1) = \prod_{\alpha \in (\mathbb{F}^* - \mathrm{QR}(\mathbb{F}^*)) \cap f^{-1}(0)} (x - \alpha)$, where $\mathrm{QR}(\mathbb{F}^*)$ is the set of squares in $\mathbb{F}^*$ and $f^{-1}(0)$ is the set of roots of f. You may use without proof the fact that $\mathbb{F}^*$ is cyclic for every finite field $\mathbb{F}$. Problem 4 on Problem Set 3 may also be helpful.

2. Explain how we can compute $\gcd(x^{(q-1)/2} - 1, f(x))$ and $\gcd(x^{(q-1)/2} + 1, f(x))$ with $\mathrm{poly}(\log q, \deg(f))$ operations over $\mathbb{F}$. (Hint: how can we compute $(x^{(q-1)/2} \pm 1) \bmod f(x)$ efficiently?)

3. Thus, by taking the gcd's of $f(x)$ with $x^{(q-1)/2} - 1$ and $x^{(q-1)/2} + 1$, we can find factors $f_1(x)$ and $f_2(x)$ of $f(x)$ that partition the non-zero roots of $f(x)$ into quadratic residues and quadratic non-residues, and thus it suffices to find the roots of $f_1(x)$ and $f_2(x)$ (by recursively applying the same method). However, this approach gets stuck if the roots of f are all quadratic residues, or all non-residues. We solve this problem by permuting the elements of $\mathbb{F}$ with a random translation $x \mapsto x + c$, which has a good chance of splitting the roots. This yields the following algorithm:

   RootFind($\mathbb{F}, f(x)$):
   Input: a description of a finite field $\mathbb{F}$ of odd order q and a nonzero polynomial $f(x) \in \mathbb{F}[x]$.
   Output: the set of roots of $f(x)$.

   (a) If $f(x)$ is of degree 0, output $\emptyset$.
   (b) If $f(x) = ax + b$ is of degree 1, output $\{-ba^{-1}\}$.
   (c) If $f(x)$ is of degree larger than 1:
      i. Choose $c \in \mathbb{F}$ uniformly at random. -
      ii. Compute $f_1(x) = \gcd((x + c)^{(q-1)/2} - 1, f(x))$ and $f_2(x) = \gcd((x + c)^{(q-1)/2} + 1, f(x))$.
      iii. Recursively compute $S_1 = \mathrm{RootFind}(\mathbb{F}, f_1(x))$ and $S_2 = \mathrm{RootFind}(\mathbb{F}, f_2(x))$.
      iv. If $f(0) = 0$, output $S_1 \cup S_2 \cup \{0\}$, else output $S_1 \cup S_2$.

   Use this algorithm to calculate all the roots in $\mathbb{F}$ of the polynomial we provide you in SAGE.