Due: Friday, 7/27, 10pm

Sundry

Before you start your homework, state briefly how you worked on it. Who else did you work with? List names and email addresses. (In case of homework party, you can just describe the group.)

1 Polynomial Practice

(a) If f and g are non-zero real polynomials, how many roots do the following polynomials have at least? How many can they have at most? (Your answer may depend on the degrees of f and g.)

- (i) $f+g$
- (ii) $f \cdot g$
- (iii) f/g, assuming that f/g is a polynomial

(b) Now let f and g be polynomials over $\mathrm{GF}(p)$.

- (i) If $f \cdot g = 0$, is it true that either $f = 0$ or $g = 0$?
- (ii) If $\deg f \geq p$, show that there exists a polynomial h with $\deg h < p$ such that $f(x) = h(x)$ for all $x \in \{0, 1, ..., p-1\}$.
- (iii) How many f of degree *exactly* $d < p$ are there such that $f(0) = a$ for some fixed $a \in \{0, 1, \ldots, p-1\}$?

(c) Find a polynomial f over $\mathrm{GF}(5)$ that satisfies $f(0) = 1, f(2) = 2, f(4) = 0$. How many such polynomials are there?

2 The CRT and Lagrange Interpolation

Let $n_1, \ldots n_k$ be pairwise coprime, i.e. n_i and n_j are coprime for all $i \neq j$. The Chinese Remainder Theorem (CRT) tells us that there exist solutions to the following system of congruences:

$$x \equiv a_1 \pmod{n_1} \tag{1}$$
$$x \equiv a_2 \pmod{n_2} \tag{2}$$
$$\vdots \tag{$\vdots$}$$
$$x \equiv a_k \pmod{n_k} \tag{k}$$

and all solutions are equivalent $\pmod{n_1 n_2 \cdots n_k}$. For this problem, parts (a)-(c) will walk us through a proof of the Chinese Remainder Theorem. We will then use the CRT to revisit Lagrange interpolation.

(a) We start by proving the $k = 2$ case: Prove that we can always find an integer x_1 that solves (1) and (2) with $a_1 = 1, a_2 = 0$. Similarly, prove that we can always find an integer x_2 that solves (1) and (2) with $a_1 = 0, a_2 = 1$.

(b) Use part (a) to prove that we can always find at least one solution to (1) and (2) for any a_1, a_2. Furthermore, prove that all possible solutions are equivalent $\pmod{n_1 n_2}$.

(c) Now we can tackle the case of arbitrary k: Use part (b) to prove that there exists a solution x to (1)-(k) and that this solution is unique $\pmod{n_1 n_2 \cdots n_k}$.

(d) For two polynomials $p(x)$ and $q(x)$, mimic the definition of $a \bmod b$ for integers to define $p(x) \bmod q(x)$. Use your definition to find $p(x) \bmod (x-1)$.

(e) Define the polynomials $x-a$ and $x-b$ to be coprime if they have no common divisor of degree 1. Assuming that the CRT still holds when replacing x, a_i and n_i with polynomials (using the definition of coprime polynomials just given), show that the system of congruences

$$p(x) \equiv y_1 \pmod{(x - x_1)} \tag{1'}$$
$$p(x) \equiv y_2 \pmod{(x - x_2)} \tag{2'}$$
$$\vdots \tag{$\vdots$}$$
$$p(x) \equiv y_k \pmod{(x - x_k)} \tag{k'}$$

has a unique solution $\pmod{(x-x_1)\cdots(x-x_k)}$ whenever the x_i are pairwise distinct. What is the connection to Lagrange interpolation?

3 Old secrets, new secrets

In order to share a secret number s, Alice distributed the values $(1, p(1)), (2, p(2)), \ldots, (n+1, p(n+1))$ of a degree n polynomial p with her friends $\text{Bob}_1, \ldots, \text{Bob}_{n+1}$. As usual, she chose p such that $p(0) = s$. Bob_1 through Bob_{n+1} now gather to jointly discover the secret. Suppose that for some reason Bob_1 already knows s, and wants to play a joke on $\text{Bob}_2, \ldots, \text{Bob}_{n+1}$, making them believe that the secret is in fact some fixed $s' \neq s$. How can he achieve this?

4 Berlekamp-Welch for General Errors

Suppose that Hector wants to send you a length $n = 3$ message, m_0, m_1, m_2, with the possibility for $k = 1$ error. For all parts of this problem, we will work mod 11, so we can encode 11 letters as shown below:

A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10

Hector encodes the message by finding the degree ≤ 2 polynomial $P(x)$ that passes through $(0, m_0)$, $(1, m_1)$, and $(2, m_2)$, and then sends you the five packets $P(0), P(1), P(2), P(3), P(4)$ over a noisy channel. The message you receive is

$$\text{DHACK} \Rightarrow 3, 7, 0, 2, 10 = r_0, r_1, r_2, r_3, r_4$$

which could have up to 1 error.

(a) First, let's locate the error, using an error-locating polynomial $E(x)$. Let $Q(x) = P(x)E(x)$. Recall that

$$Q(i) = P(i)E(i) = r_i E(i), \quad \text{for} \quad 0 \leq i < n + 2k.$$

What is the degree of $E(x)$? What is the degree of $Q(x)$? Using the relation above, write out the form of $E(x)$ and $Q(x)$ in terms of the unknown coefficients, and then a system of equations to find both these polynomials.

(b) Solve for $Q(x)$ and $E(x)$. Where is the error located?

(c) Finally, what is $P(x)$? Use $P(x)$ to determine the original message that Hector wanted to send.

5 Error-Detecting Codes

Suppose Alice wants to transmit a message of n symbols, so that Bob is able to *detect* rather than *correct* any errors that have occured on the way. That is, Alice wants to find an encoding so that Bob, upon receiving the code, is able to either

(I) tell that there are no errors and decode the message, or

(II) realize that the transmitted code contains at least one error, and throw away the message.

Assuming that we are guaranteed a maximum of k errors, how should Alice extend her message (i.e. by how many symbols should she extend the message, and how should she choose these symbols)? You may assume that we work in GF(p) for very large prime p. Show that your scheme works, and that adding any lesser number of symbols is not good enough.