

Discrete Mathematics: Homework #5

Due on March 1, 2025 at 6:00pm

Professor Satish Rao

Zachary Brandt
zbrandt@berkeley.edu

Problem 1: Equivalent Polynomials

This problem is about polynomials with coefficients in $\text{GF}(p)$ for some prime $p \in \mathbb{N}$. We say that two such polynomials f and g are *equivalent* if $f(x) \equiv g(x) \pmod{p}$ for every $x \in \text{GF}(p)$.

- A) Show that $f(x) = x^{p-1}$ and $g(x) = 1$ are **not** equivalent polynomials under $\text{GF}(p)$.
- B) Use Fermat's Little Theorem to find a polynomial with degree strictly less than 5 that is equivalent to $f(x) = x^5$ over $\text{GF}(5)$; then find a polynomial with degree strictly less than 11 that is equivalent to $g(x) = 4x^{70} + 9x^{11} + 3$ over $\text{GF}(11)$.
- C) In $\text{GF}(p)$, prove that whenever $f(x)$ has degree $\geq p$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< p$.

Problem 2: Secret Sharing

Suppose the Oral Exam questions are created by 2 TAs and 3 Readers. The answers are all encrypted, and we know that:

- A) Two TAs together should be able to access the answers
- B) Three Readers together should be able to access the answers
- C) One TA and one Reader together should also be able to access the answers
- D) One TA by themselves or two Readers by themselves should not be able to access the answers.

Design a Secret Sharing scheme to make this work.

Problem 3: To The Moon!

A secret number s is required to launch a rocket, and Alice distributed the values $(1, p(1)), (2, p(2)), \dots, (n+1, p(n+1))$ of a degree n polynomial $p(x)$ to a group of $n+1$ GME holders $\text{Bob}_1, \dots, \text{Bob}_{n+1}$. As usual, she chose $p(x)$ such that $p(0) = s$. Bob_1 through Bob_{n+1} now gather to jointly discover the secret. However, Bob_1 is secretly a partner at Melvin Capital and already knows s , and wants to sabotage $\text{Bob}_2, \dots, \text{Bob}_{n+1}$, making them believe that the secret is in fact some fixed $s' \neq s$. How could he achieve this? In other words, what value should he report (in terms variables known in the problem, such as s', s or $p(1)$) in order to make the others believe that the secret is s' ?

Problem 4: Lagrange? More like Lamegrangle.

- A) Let's say we wanted to interpolate a polynomial through a single point, (x_0, y_0) . What would be the polynomial that we would get? (This is not a trick question. A degree 0 polynomial is fine.)
- B) Call the polynomial from the previous part $f_0(x)$. Now say we wanted to define the polynomial $f_1(x)$ that passes through the points (x_0, y_0) and (x_1, y_1) . If we write $f_1(x) = f_0(x) + a_1(x - x_0)$, what value of a_1 causes $f_1(x)$ to pass through the desired points?
- C) Now say we want a polynomial $f_2(x)$ that passes through (x_0, y_0) , (x_1, y_1) , and (x_2, y_2) . If we write $f_2(x) = f_1(x) + a_2(x - x_0)(x - x_1)$, what value of a_2 gives us the desired polynomial?
- D) Suppose we have a polynomial $f_i(x)$ that passes through the points (x_0, y_0) , ..., (x_i, y_i) and we want to find a polynomial $f_{i+1}(x)$ that passes through all those points and also (x_{i+1}, y_{i+1}) . If we define $f_{i+1}(x) = f_i(x) + a_{i+1} \prod_{j=0}^i (x - x_j)$, what value must a_{i+1} take on?

Problem 5: Error-Correcting Codes

- A) Recall from class the error-correcting code for erasure errors, which protects against up to k lost packets by sending a total of $n + k$ packets (where n is the number of packets in the original message). Often the number of packets lost is not some fixed number k , but rather a *fraction* of the number of packets sent. Suppose we wish to protect against a fraction α of lost packets (where $0 < \alpha < 1$). At least how many packets do we need to send (as a function of n and α)?
- B) Repeat part (A) for the case of general errors.

Problem 6: Alice and Bob

- A) Alice decides that instead of encoding her message as the values of a polynomial, she will encode her message as the coefficients of a degree 2 polynomial $P(x)$. For her message $[m_1, m_2, m_3]$, she creates the polynomial $P(x) = m_1x^2 + m_2x + m_3$ and sends the five packets $(0, P(0))$, $(1, P(1))$, $(2, P(2))$, $(3, P(3))$, and $(4, P(4))$ to Bob. However, one of the packet y -values (one of the $P(i)$ terms; the second attribute in the pair) is changed by Eve before it reaches Bob. If Bob receives

$$(0, 1), (1, 3), (2, 0), (3, 1), (4, 0)$$

and knows Alice's encoding scheme and that Eve changed one of the packets, can he recover the original message? If so, find it as well as the x -value of the packet that Eve changed. If he can't, explain why. Work in mod 7. Also, feel free to use a calculator or online systems of equations solver, but make sure it can work under mod 7.

- B) Bob gets tired of decoding degree 2 polynomials. He convinces Alice to encode her messages on a degree 1 polynomial. Alice, just to be safe, continues to send 5 points on her polynomial even though it is only degree 1. She makes sure to choose her message so that it can be encoded on a degree 1 polynomial. However, Eve changes two of the packets. Bob receives $(0, 5)$, $(1, 7)$, $(2, x)$, $(3, 5)$, $(4, 0)$. If Alice sent $(0, 5)$, $(1, 7)$, $(2, 9)$, $(3, -2)$, $(4, 0)$, for what values of x will Bob not uniquely be able to determine Alice's message? Assume that Bob knows Eve changed two packets. Work in mod 13. Again, feel free to use a calculator or graphing calculator software.

Hint: Observe that since Bob knows that Eve changed two packets, he's looking for a polynomial that passes through at least 3 of the given points. Think about what must happen in order for Bob to be unable to uniquely identify the original polynomial.

- C) Alice wants to send a length n message to Bob. There are two communication channels available to her: Channel X and Channel Y. Only 6 packets can be sent through channel X. Similarly, Channel Y will only deliver 6 packets, but it will also corrupt (change the value) of one of the delivered packets. Using each of the two channels once, what is the largest message length n such that Bob so that he can always reconstruct the message?