CS 70 Discrete Mathematics and Probability Theory
Spring 2022 Koushik Sen and Satish Rao **HW 05**

1 Lagrange? More like Lamegrange.

In this problem, we walk you through an alternative to Lagrange interpolation.

(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.)

(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?

Solution:

(a) We want a degree zero polynomial, which is just a constant function. The only constant function that passes through (x_0, y_0) is $f_0(x) = y_0$.

(b) By defining $f_1(x) = f_0(x) + a_1(x - x_0)$, we get that

$$f_1(x_0) = f_0(x_0) + a_1(x_0 - x_0) = y_0 + 0 = y_0.$$

So now we just need to make sure that $f_1(x_1) = y_1$. This means that we need to choose a_1 such that

$$f_1(x_1) = f_0(x_1) + a_1(x_1 - x_0) = y_1.$$

Solving this for a_1, we get that

$$a_1 = \frac{y_1 - f_0(x_1)}{x_1 - x_0}.$$

(c) We apply similar logic to the previous part. From our definition, we know that

$$f_2(x_0) = f_1(x_0) + a_2(x_0 - x_0)(x_0 - x_1) = y_0 + 0 = y_0.$$

and that

$$f_2(x_1) = f_1(x_1) + a_2(x_1 - x_0)(x_1 - x_1) = y_1 + 0 = y_1.$$

Thus, we just need to choose a_2 such that $f_2(x_2) = y_2$. Putting in our formula for $f_2(x)$, we get that we need a_2 such that

$$f_1(x_2) + a_2(x_2 - x_0)(x_2 - x_1) = y_2.$$

Solving for a_2, we get that

$$a_2 = \frac{y_2 - f_1(x_2)}{(x_2 - x_0)(x_2 - x_1)}.$$

(d) If we try to calculate $f_{i+1}(x_k)$ for $0 \le k \le i$, we know one of the $(x - x_j)$ terms (specifically the kth one) will be zero. Thus, we get that

$$f_{i+1}(x_k) = f_i(x_k) + a_{i+1}(0) = y_k + 0 = y_k.$$

So now we just need to pick a_i such that $f_{i+1}(x_{i+1}) = y_{i+1}$. This means that we need to choose a_{i+1} such that

$$f_i(x_{i+1}) + a_{i+1} \prod_{j=0}^{i} (x_{i+1} - x_j) = y_{i+1}.$$

Solving for a_{i+1}, we get that

$$a_{i+1} = \frac{y_{i+1} - f_i(x_{i+1})}{\prod_{j=0}^{i}(x_{i+1} - x_j)}.$$

The method you derived in this question is known as Newtonian interpolation. (The formal definition of Newtonian interpolation uses divided differences, which we don't cover in this class, but it's in effect doing the same thing.) This method has an advantage over Lagrange interpolation in that it is very easy to add in extra points that your polynomial has to go through (as we showed in part (c), whereas Lagrange interpolation would require you to throw out all your previous work and restart. However, if you want to keep the same x values but change the y values, Newtonian interpolation requires you to throw out all your previous work and restart. In contrast, this is fairly easy to do with Lagrange interpolation–since changing the y values doesn't affect the δ_is, you don't have to recalculate those, so you can skip most of the work.

2 Equivalent Polynomials

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

(a) Use Fermat's Little Theorem to find a polynomial with degree strictly less than 5 that is equivalent to $f(x) = x^5$ over GF(5); then find a polynomial with degree strictly less than 11 that is equivalent to $g(x) = 1 + 3x^{11} + 7x^{13}$ over GF(11).

(b) Prove that whenever $f(x)$ has degree $\ge q$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< q$.

Solution:

(a) Fermat's Little Theorem says that for any nonzero integer a and any prime number q, $a^{q-1} \equiv 1 \mod q$. We're allowed to multiply through by a, so the theorem is equivalent to saying that $a^q \equiv a \mod q$; note that this is true even when $a = 0$, since in that case we just have $0^q \equiv 0 \mod q$. The problem asks for a polynomial $\tilde{f}(x)$, different from $f(x)$, with the property that $\tilde{f}(a) \equiv a^5 \mod 5$ for any integer a. Directly using the theorem, $\tilde{f}(x) = x$ will work. We can do something similar with $g(x) = 1 + 3x^{11} + 7x^{13}$ modulo 11: set $\tilde{g}(x) = 1 + 3x + 7x^3$.

(b) One proof uses Fermat's Little Theorem. As a warm-up, let $d \geq q$; we'll find a polynomial equivalent to x^d. For any integer, we know

$$\begin{aligned} a^d &= a^{d-q}a^q \\ &\equiv a^{d-q}a \mod q \\ &\equiv a^{d-q+1} \mod q. \end{aligned}$$

In other words x^d is equivalent to the polynomial $x^{d-(q-1)}$. If $d-(q-1) \geq q$, we can show in the same way that x^d is equivalent to $x^{d-2(q-1)}$. Since we subtract $q-1$ every time, the sequene $d, d-(q-1), d-2(q-1), \ldots$ must eventually be smaller than q. Now if $f(x)$ is any polynomial with degree $\geq q$, we can apply this same trick to every x^k that appears for which $k \geq q$.

Another proof uses Lagrange interpolation. Let $f(x)$ have degree $\geq q$. By Lagrange interpolation, there is a unique polynomial $\tilde{f}(x)$ of degree at most $q-1$ passing through the points $(0, f(0)), (1, f(1)), (2, f(2)), \ldots, (q-1, f(q-1))$, and we designed it exactly so that it would be equivalent to $f(x)$.

3 The CRT and Lagrange Interpolation

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

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} && (1) \\ x &\equiv a_2 \pmod{n_2} && (2) \\ &\vdots && (\vdots) \\ x &\equiv a_k \pmod{n_k} && (k) \end{aligned}$$

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 polynomials $p_1(x)$, $p_2(x)$ and $q(x)$ we say that $p_1(x) \equiv p_2(x) \bmod q(x)$ if $p_1(x) - p_2(x)$ is of the form $q(x) \times m(x)$ for some polynomial $m(x)$.

Define the polynomials $x - a$ and $x - b$ to be co-prime 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 co-prime polynomials just given), show that the system of congruences

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

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?

Hint: To show that a unique solution exists, you may use the fact that the CRT has a unique solution when certain properties are satisfied.

Solution:

(a) Since $\gcd(n_1, n_2) = 1$, there exist integers k_1, k_2 such that $1 = k_1 n_1 + k_2 n_2$. Setting $x_1 = k_2 n_2 = 1 - k_1 n_1$ and $x_2 = k_1 n_1 = 1 - k_2 n_2$ we obtain the two desired solutions.

(b) Using the x_1 and x_2 we found in Part (a), we show that $a_1 x_1 + a_2 x_2 \pmod{n_1 n_2}$ is a solution to the desired equivalences:

$$\begin{aligned} a_1 x_1 + a_2 x_2 &\equiv a_1 \cdot 1 + a_2 \cdot 0 \equiv a_1 \pmod{n_1} \\ a_1 x_1 + a_2 x_2 &\equiv a_1 \cdot 0 + a_2 \cdot 1 \equiv a_2 \pmod{n_2}. \end{aligned}$$

Such result is also unique. Say that we have two difference solutions $x = c$ and $x = c'$, which both satisfy $x \equiv a_1 \pmod{n_1}$ and $x \equiv a_2 \pmod{n_2}$. This would give us $c \equiv c' \pmod{n_1}$ and $c \equiv c' \pmod{n_2}$, which suggests that $(c - c')$ is divisible by n_1 and n_2. Since n_1 and n_2 are coprime, $gcd(n_1, n_2) = 1$, $(c - c')$ is divisible by $n_1 n_2$. Writing it in modular form gives us $c \equiv c' \pmod{n_1 n_2}$. Therefore, all the result is unique with respect to $\pmod{n_1 n_2}$

(c) We use induction on k. Part (b) handles the base case, $k = 2$. For the inductive hypothesis, assume for $k \leq l$, the system (1)-(k) has a unique solution $a \pmod{n_1 \cdots n_k}$. Now consider $k = l + 1$, so we add the equation $x \equiv a_{l+1} \pmod{n_{l+1}}$ to our system, resulting in

$$\begin{aligned} x &\equiv a \pmod{n_1 \cdots n_l} \\ x &\equiv a_{l+1} \pmod{n_{l+1}}. \end{aligned}$$

Since the n_i are pairwise coprime, $n_1 n_2 \cdots n_l$ and n_{l+1} are coprime. Part (b) tells us that there exists a unique solution $a' \pmod{n_1 \cdots n_l n_{l+1}}$. We conclude that a' is the unique solution to (1)-($l+1$), when taken $\pmod{n_1 n_2 \cdots n_l n_{l+1}}$.

(d) We only need to check that $q_i(x) = (x - x_i)$ and $q_j(x) = (x - x_j)$ are coprime whenever $x_i \neq x_j$; that is, that they don't share a common divisor of degree 1. If $d_i(x) = a_i x + b_i$ is a divisor of $q_i(x)$, then $q_i(x) = q'(x)(a_i x + b_i)$ for some polynomial $q'(x)$. But since $q_i(x)$ is of degree 1, $q'(x)$ must be of degree 0 and hence a constant, so $d_i(x)$ must be a constant multiple of $q_i(x)$. Similarly, any degree 1 divisor d_j of $q_j(x)$ must be a constant multiple of $q_j(x)$, and if $x_i \neq x_j$, then none of these multiples overlap, so $q_i(x)$ and $q_j(x)$ are coprime.

From our result in part (d), the congruences ($1'$)-(k') assert that we are looking for a polynomial $p(x)$ such that $p(x_i) = y_i$. The CRT then establishes the existence of $p(x)$, and that it is unique modulo a degree k polynomial. That is, $p(x)$ is unique if its degree is at most $k-1$. Lagrange interpolation finds $p(x)$.

4 One Point Interpolation

Suppose we have a polynomial $f(x) = x^k + c_{k-1}x^{k-1} + \cdots + c_2 x^2 + c_1 x + c_0$.

(a) Can we determine $f(x)$ with k points? If so, provide a set of inputs $x_0, x_1, \ldots, x_{k-1}$ such that knowing points $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_{k-1}, f(x_{k-1}))$ allows us to uniquely determine $f(x)$, and show how $f(x)$ can be determined from such points. If not, provide a proof of why this is not possible.

(b) Now, assume each coefficient is an integer satisfying $0 \leq c_i < 100 \quad \forall i \in [0, k-1]$. Can we determine $f(x)$ with one point? If so, provide an input x_* such that knowing the point $(x_*, f(x_*))$ allows us to uniquely determine $f(x)$, and show how $f(x)$ can be determined from this point. If not, provide a proof of why this is not possible.

Solution:

(a) Yes. Since the leading coefficient is 1, we only need to find the k remaining coefficients $c_0, c_1, \ldots, c_{k-1}$ to determine $f(x)$. This can be done with *any* k distinct points.

For example, suppose we know the points $(0, f(0)), (1, f(1)), \ldots, (k-1, f(k-1))$. We can then write the degree $k-1$ polynomial

$$g(x) = c_{k-1}x^{k-1} + \cdots + c_2 x^2 + c_1 x + c_0 = f(x) - x^k$$

which can be determined via Lagrange interpolation on $(0, f(0)), (1, f(1) - 1), (2, f(2) - 2^k), \ldots, (k-1, f(k-1) - (k-1)^k)$, uniquely yielding our desired coefficients $c_0, c_1, \ldots, c_{k-1}$.

(b) Yes. We can express each nonnegative two-digit integer $c_i = 10d_{2i+1} + d_{2i}$ for digits $d_i \in [0, 9]$.

Using $x_* = 100$,

$$\begin{aligned} f(100) &= 100^k + c_{k-1}100^{k-1} + \cdots + c_2 100^2 + c_1 100 + c_0 \\ &= 10^{2k} + (10d_{2k-1} + d_{2k-2})10^{2k-2} + \cdots + (10d_5 + d_4)10^4 + (10d_3 + d_2)10^2 + (10d_1 + d_0) \\ &= 10^{2k} + 10^{2k-1}d_{2k-1} + 10^{2k-2}d_{2k-2} + \cdots + 10^5 d_5 + 10^4 d_4 + 10^3 d_3 + 10^2 d_2 + 10d_1 + d_0 \end{aligned}$$

Thus, the rightmost $2k-1$ digits of $f(100)$, from right to left, are $d_0, d_1, \ldots, d_{2k-1}$; we can then determine our desired coefficients $c_i = 10d_{2i+1} + d_{2i}$.

5 Secret Sharing

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

- Two TAs together should be able to access the answers
- Three Readers together should be able to access the answers
- One TA and one Reader together should also be able to access the answers

Design a Secret Sharing scheme to make this work.

Solution:

Solution 1 We can use a degree 2 polynomial, which is uniquely determined by 3 points. Evaluate the polynomial at 7 points, and distribute a point to each Reader and 2 points to each TA. Then, all possible combinations will have at least 3 points to recover the answer key.

Basically, the point of this problem is to assign different weight to different class of people. If we give one share to everyone, then 2 Readers can also recover the secret and the scheme is broken.

Solution 2 We construct three polynomials, one for each way of recovering the answer key:

- A degree 1 polynomial for recovering with two TAs, evaluated at 2 points. Distribute a point to each TA.
- A degree 2 polynomial for recovering with three readers, evaluated at 3 points. Distribute a point to each Reader.
- A degree 1 polynomial for recovering with one TA + one reader. Evaluate this polynomial at 2 points, and distribute one point to all TAs and one point to all readers.

All combinations can then use the corresponding polynomial to recover the answer key.

6 Trust No One

Gandalf has assembled a fellowship of eight peoples to transport the One Ring to the fires of Mount Doom: four hobbits, two humans, one elf, and one dwarf. The ring has great power that may be of use to the fellowship during their long and dangerous journey. Unfortunately, the use of its immense power will eventually corrupt the user, so it must not be used except in the most dire of circumstances. To safeguard against this possibility, Gandalf wishes to keep the instructions a secret from members of the fellowship. The secret must only be revealed if enough members of the fellowship are present and agree to use it.

Gandalf has hired your services to help him come up with a secret sharing scheme that accomplishes this task, summarized by the following points:

- There is a party of four hobbits, two humans, an elf, and a dwarf, and a secret message that must remain unknown to everyone if not enough members of the party agree.
- A group of people consisting of at least two people from different people classes and at least one people class that is fully represented (i.e., has all members present) can unlock the secret of the ring.

A few examples: only four hobbits agreeing to use the ring is not enough to know the instructions. One human and three hobbits is not enough. However, all four hobbits and one human agreeing is enough. Both humans and the dwarf agreeing is enough.

Solution:

Solution 1

There will be two parts to this secret: a unanimity secret U and a multi-people secret M. U ensures that at least all members of one peoples are in agreement while M ensures that members of at least two peoples are in agreement.

The high-level idea is that the secret of the ring requires both the unanimity and multi-people conditions to be satisfied, so we encode the original secret in a polynomial $R(x)$ determinable by the two values U and M; each of U and M themselves are encoded within polynomials as independent secrets determinable when the unanimity and multi-people conditions, respectively, are satisfied. Thus, once both U and M are recovered, they can then be combined to reveal the original secret, since each will be a point of the degree-1 polynomial $R(x)$ whose y-intercept contains the secret of the ring.

We will now detail U and M in order below.

The *unanimity secret* involves creating a separate secret for each people. We will require all members of that people to join forces in order to reveal the secret. For example, the hobbits will each have distinct points of a degree-3 polynomial and the humans will each have distinct points of a degree-1 polynomial. When all members of a people come together, they will reveal U (encoded, for example, as the y-intercept of each of these polynomials). Note that the elf and the dwarf each know U already since they are the only members of their people.

The *multi-people secret* involves creating a degree-1 polynomial $P_m(x)$ and giving one point to all members of each people. For example, the hobbits may each get $P_m(1)$ while the elf gets $P_m(2)$ and the humans each get $P_m(3)$. In this way if members of any two peoples are in agreement, they can reveal M (encoded, for example, as the y-intercept of $P_m(x)$).

Once U and M are each known, they can be *combined* to determine the final secret. U and M allow us to uniquely determine $R(x)$ and thus $R(0)$, the secret of the ring.

This scheme is an example of hierarchical secret sharing. Let's work out a specific example.

Example: Suppose the secret is $s = 4$, $M = 3$, and $U = 2$. From now on, we can work in GF(5) since $s < 5$ and $n < 5$ (n is the number of people who have pieces of the secret).

First we need to create a degree-1 polynomial $R(x)$ such that $R(0) = s = 4$, $R(1) = M = 3$, and $R(2) = U = 2$. By inspection, $R(x) = 4x + 4$ has these properties (e.g. $R(1) = 4 \cdot 1 + 4 = 8 \equiv 3$).

Now we can create the multi-people secret M. We choose degree-1 polynomial $P_m(x) = x + 3$ and tell each hobbit $P_m(1) = 4$, the elf $P_m(2) = 5 \equiv 0$, each of the humans $P_m(3) = 6 \equiv 1$, and the dwarf $P_m(4) = 7 \equiv 2$. Now any two members of distinct peoples can determine $P_m(x)$ and thus $P_m(0)$ by interpolating their two values.

When creating the unanimity secret U, we first note that each of the dwarf and the elf will be told U directly since they are the only members of their respective people. On the other hand, the humans will each have a point on the degree-1 polynomial $P_{humans}(x)$. Suppose $P_{humans}(x) = 2x + 2$. Then the first human receives $P_{humans}(1) = 4$ and the second receives $P_{humans}(2) = 4 + 2 = 6 \equiv 1$. When they interpolate using these values, they will discover the original polynomial and therefore $P_{humans}(0) = U = 2$. The hobbits will have a similar secret but with a degree-3 polynomial (e.g. $P_{hobbit}(x) = 4x^3 + x^2 + 2$).

Now suppose that two humans and one hobbit come together. The two humans work together to determine U as described above. Together the three of them also know $P_m(3) = 6$ and $P_m(1) = 4$, from which they can find $P_m(x)$ and thus $P_m(0) = M = 3$. Now that they have U and M, they can interpolate to find $R(x)$ and thus $R(0) = s = 4$.

Solution 2

Alternatively, we can construct a single degree 6 polynomial and distribute 1 point to each hobbit, 3 points to each human, 6 points to the elf, and 6 points to the dwarf. We can see that if all the hobbits agree, they will need 3 more points in order to interpolate successfully and each member of all the other peoples are given at least 3 points. Moreover, each of the other peoples have 6 points in total, meaning that if all the humans, the elf, or the dwarf agree, they'll only need one more point which can be provided by any additional member of the party outside their people. On the other hand, the most amount of points that could be obtained from an agreeing group that does not satisfy the requirements would be 6, from the group consisting of one human and all the hobbits. This would be insufficient to interpolate the polynomial so therefore, the scheme fulfills the requirements.

7 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.

Solution:

(a) Suppose we send a total of m packets (where m is to be determined). Since at most a fraction α of these are lost, the number of packets received is at least $(1-\alpha)m$. But in order to reconstruct the polynomial used in transmission, we need at least n packets. Hence it is sufficient to have $(1-\alpha)m \geq n$, which can be rearranged to give $m \geq n/(1-\alpha)$.

(b) Suppose we send a total of $m = n+2k$ packets, where k is the number of errors we can guard against. The number of corrupted packets is at most αm, so we need $k \geq \alpha m$. Hence $m \geq n+2\alpha m$. Rearranging gives $m \geq n/(1-2\alpha)$.

Note: Recovery in this case is impossible if $\alpha \geq 1/2$.

8 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 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.

(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.

(c) Alice wants to send a length 9 message to Bob. There are two communication channels available to her: Channel A and Channel B. When n packets are fed through Channel A, only 6 packets, picked arbitrarily, are delivered. Similarly, Channel B will only deliver 6 packets, picked arbitrarily, but it will also corrupt (change the value) of one of the delivered packets. Each channel will only work if at least 10 packets are sent through it. Using each of the two channels once, provide a way for Alice to send her message to Bob so that he can always reconstruct it.

Solution:

(a) We can use Berlekamp and Welch. We have: $Q(x) = P(x)E(x)$. $E(x)$ has degree 1 since we know we have at most 1 error. $Q(x)$ is degree 3 since $P(x)$ is degree 2. We can write a system of linear equations and solve:

$$\begin{aligned} d &= 1(0-e) \\ a+b+c+d &= 3(1-e) \\ 8a+4b+2c+d &= 0(2-e) \\ 27a+9b+3c+d &= 1(3-e) \\ 64a+16b+4c+d &= 0(4-e) \end{aligned}$$

Since we are working in mod 7, this is equivalent to:

$$\begin{aligned} d &= -e \\ a+b+c+d &= 3-3e \\ a+4b+2c+d &= 0 \\ 6a+2b+3c+d &= 3-e \\ a+2b+4c+d &= 0 \end{aligned}$$

Solving yields:

$$Q(x) = x^3+5x^2+5x+4, E(x) = x-3$$

To find $P(x)$ we divide $Q(x)$ by $E(x)$ and get $P(x) = x^2+x+1$. So Alice's message is $m_1 = 1, m_2 = 1, m_3 = 1$. The x-value of the packet Eve changed is 3.

<u>Alternative solution</u>: Since we have 5 points, we have to find a polynomial of degree 2 that goes through 4 of those points. The point that the polynomial does not go through will be the packet that Eve changed. Since 3 points uniquely determine a polynomial of degree 2, we can pick 3 points and check if a 4th point goes through it. (It may be the case that we need to try all sets of 3 points.) We pick the points $(1,3),(2,0),(4,0)$. Lagrange interpolation can be used to create the polynomial but we can see that for the polynomial that goes through these 3 points, it has 0s at $x=2$ and $x=4$. Thus the polynomial is $k(x-2)(x-4) = k(x^2-6x+8) \pmod 7 \equiv k(x^2+x+1) \pmod 7$. We find $k \equiv 1$ by plugging in the point $(1,3)$, so our polynomial is x^2+x+1. We then check to see if the this polynomial goes through one of the 2 points that we didn't use. Plugging in 0 for x, we get 1. The packet that Eve changed is the point that our polynomial does not go through which has x-value 3. Alice's original message was $m_1 = 1, m_2 = 1, m_3 = 1$.

(b) Since Bob knows that Eve changed 2 of the points, the 3 remaining points will still be on the degree 1 polynomial that Alice encoded her message on. Thus if Bob can find a degree 1 polynomial that passes through at least 3 of the points that he receives, he will be able to uniquely recover Eve's message. The only time that Bob cannot uniquely determine Alice's message is if there are 2 polynomials with degree 1 that pass through 3 of the 5 points that he receives. Since we are working with degree 1 polynomials, we can plot the points that Bob receives and then see which values of x will cause 2 sets of 3 points to fall on a line. $(0,5),(1,7),(4,0)$ already fall on a line. If $x = 6$, $(1,7),(2,6),(3,5)$ also falls on a line. If $x = 5$, $(0,5),(2,5),(3,5)$ also falls on a line. If $x = 9$, $(0,5),(2,9),(4,0)$ falls on the original line, so here Bob can decode the message. If $x = 10$, $(2,10),(3,5),(4,0)$ also falls on a line. So if $x = 6,5,10$, Bob will not be able to uniquely determine Alice's message.

(c) Channel A will deliver 6 packets so we can send a message of length 6 encoded on a polynomial of degree 5 though it. If we send 10 points though channel A, it doesn't matter which 6 points Bob gets, he will still be able to reconstruct our degree 5 polynomial. Since the channel B has 1 general error, we can only send a message of length 4 encoded on a degree 3 polynomial through it. If we send 10 points, Bob will get 6 points to calculate a degree 4 polynomial with 1 general error, which he is able to do. Thus to send our length 8 message, we can send the character 1 - 6 through a channel A and the characters 7 - 9 through channel B.

Alternative Solution: Alice can interpolate a polynomial of degree 8 encoding the message of length 9. She sends 10 points from that polynomial through channel A and another 10 points from the same polynomial through channel B. Bob will receive 6 points from channel A and 6 points from channel B, with one of them corrupted. He can use Berlekamp-Welch with $n = 9$ and $k = 1$ to recover the original polynomial. He retrieves the message by evaluating the polynomial on relevant points.