CMIMC 2020

Power Round

INSTRUCTIONS

1. Do not look at the test before the proctor starts the round.
2. This test consists of several problems, some of which are short-answer and some of which require proofs, to be solved within a time frame of **60 minutes**. There are **70 points** total.
3. Answers should be written and clearly labeled on sheets of blank paper. Each numbered problem should be *on its own sheet*. If you have multiple pages, number them as well (e.g. 1/3, 2/3).
4. Write your team ID on the upper-right corner and the problem and page number of the problem whose solution you are writing on the upper-left corner on each page you submit. Papers missing these will not be graded. Problems with more than one submission will not be graded.
5. Write legibly. Illegible handwriting will not be graded.
6. In your solution for any given problem, you may assume the results of previous problems, even if you have not solved them. You may not do the same for later problems.
7. Problems are not ordered by difficulty. They are ordered by progression of content.
8. No computational aids other than pencil/pen are permitted.
9. If you believe that the test contains an error, submit your protest in writing to Doherty 2302 by the end of lunch.

CMIMC 2020

In this power round, we will explore the mathematics of Coding Theory.

Contents

1 Introduction

Alice has moved to a new rent-controlled condo two thousand feet under the sea. Unfortunately, whenever she sends letters to her pen-pal Bob, the water dampens the letters and makes them hard to read. Specifically, their letters can be corrupted in two different ways:

- **Erasures:** The water obscures symbols in the transmitted messages. This can be understood as replacing them with '?'s.
- **Errors:** The water turns some of the symbols into other symbols.

Is it possible for Bob to error-proof his messages so that they can be decoded even if the water corrupts them? This simple question is the basis of "coding theory", which is an important and active area of computer science research today. In this power round, we will explore the mathematics of sending messages and apply them to some novel situations.

Formally, we define a message as follows:

Definition 1 (Alphabet)**.** An *alphabet* is a finite set of symbols Σ, such as $\{0, 1\}$ or $\{a, b, c\}$.

Definition 2 (Message)**.** A *message* is a sequence of symbols, selected from some alphabet Σ. The set of messages of length k over Σ is indicated as Σ^k. For example, $\{a, b\}^2 = \{aa, ab, ba, bb\}$.

For example, if Alice wants to send Bob the message "GOOD LUCK," the message with an erasure would be "GOO? LUCK," and the message with an error might be "GOOF LUCK."

2 Some Elementary Codes [18 points]

In this section, we shall devise some simple strategies to help Alice send messages to Bob in a way that allows him to detect and correct these two types of errors.

Definition 3 (Code)**.** A *(block) code* $\mathcal{C}$ is a subset of Σ^n, where n is called the *block length.* Elements of a code are called *codewords.*

Definition 4 (Encoding Map)**.** An *encoding map* $E : \Sigma^k \to \Sigma^n$ is an injective function from message set $\mathcal{M} = \Sigma^k$ to Σ^n (i.e. no two words in Σ^k map to the same word under E). The resultant code is the image of the map, i.e. $\mathcal{C} = E(\mathcal{M})$. The integer k is called the *message length.*

In order to send a message $m \in \mathcal{M}$ using a code, we instead transmit $E(m)$ over the channel. Thus, the goal is to devise codes and encoding maps so that we are able to decode m from a "corrupted" version of $E(m)$.

Problem 2.1 (Repetition Code, 2 points)

A simple way of handling corruption is to encode m by repeating each symbol of message m a certain number of times.

(i) (1 point) Suppose the channel can introduce up to t erasures. How many times should we repeat each symbol in the message to ensure that we can recover the original message?

(ii) (1 point) Repeat the previous problem, now assuming the channel can introduce up to t errors.

Definition 5 (q-ary Codes)**.** A code over the alphabet $\Sigma_q = \{0, 1, \ldots, q-1\}$ is called a *q-ary* code. A 2-ary code is called a *binary* code.

Problem 2.2 (2 points)

Devise a code/encoding map with the smallest possible block length for sending the message set Σ_2^k across a channel which can introduce at most one erasure and no errors.

Problem 2.3 (6 points)

Devise a code/encoding map for sending the message set Σ_2^k across a channel which can introduce at most one error and no erasures. Your code should have block length $n \leq 2k+1$. You may assume that k is sufficiently large. Points will be awarded based on how small the block length is.

While the codes above are easy solutions, they are not very efficient and thus not good solutions to Alice's problem. We now move toward a family of codes that are much more efficient (and actually implemented in practice!). To this end, we need a few definitions.

Definition 6 (Hamming Distance)**.** For two messages of equal length x and y, the *Hamming distance* $d(x, y)$ is the number of indices in the string where the messages differ. For example, $d(00221, 01011) = 3$.

Definition 7 (Hamming Weight)**.** For a string m, the *Hamming Weight* $\mathrm{wt}(m)$ is the number of symbols which are nonzero. For example, $\mathrm{wt}(012001) = 3$.

Definition 8 (Minimum Distance)**.** The *minimum distance* of a code $\mathcal{C}$ is defined as the smallest hamming distance between any two distinct codewords in $\mathcal{C}$. Formally,

$$d(\mathcal{C}) = \min_{x \neq y \in \mathcal{C}} d(x, y)$$

We also need the notion of a *linear code.*

Definition 9 (Linear Code)**.** A q-ary encoding map $E : \Sigma_q^k \to \Sigma_q^n$ is said to be linear if

- $E(0) = 0$
- For all $x, y \in \Sigma_q^k$, $E(x+y) = E(x) + E(y)$

where addition is element-wise modulo q. The resultant code is called a linear code.

Problem 2.4 (2 points)

Prove that for any q-ary linear code $\mathcal{C}$, the following relation holds.

$$d(\mathcal{C}) = \min_{c \in \mathcal{C}, c \neq 0} \mathrm{wt}(c)$$

Problem 2.5 (4 points)

Prove that the following statements for a code $\mathcal{C}$ are equivalent.

1. $\mathcal{C}$ has minimum distance at least $2t+1$.
2. $\mathcal{C}$ can be used to correct t symbol errors.
3. $\mathcal{C}$ can be used to correct $2t$ symbol erasures.

Problem 2.6 (2 points)

Let the (7, 4) **Hamming code** be the 2-ary code obtained by encoding map

$$(x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3, x_4, x_1 \oplus x_2 \oplus x_4, x_1 \oplus x_3 \oplus x_4, x_2 \oplus x_3 \oplus x_4)$$

where $\oplus$ denotes addition modulo 2. Find, with proof, the minimal distance of this code.

Remark. The above construction can be generalized for larger values of 7 and 4.

3 Bounds on Code Size [12 points]

In devising codes, there is always a trade-off between message length (how much we can transmit) and code distance (how much we can correct). In this section, we will derive a few bounds quantifying this trade-off.

Problem 3.1 (Singleton Bound, 3 points)

Let $\mathcal{C}$ be a q-ary code with block length n and minimum distance d. Prove that

$$|\mathcal{C}| \leq q^{n-d+1}.$$

Problem 3.2 (4 points)

Let $\mathcal{C}$ be a binary code with block length n and minimum distance 3. Prove that

$$|\mathcal{C}| \leq \frac{2^n}{n+1}.$$

Problem 3.3 (Hamming Bound, 3 points)

Let $\mathcal{C}$ be a q-ary code with block length n and minimum distance d. Generalize the previous problem to prove that, for $t = \lfloor \frac{d-1}{2} \rfloor$,

$$|\mathcal{C}| \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i}.$$

Problem 3.4 (2 points)

Using the previous problem, prove that, for any $1 \leq d \leq k$, any binary encoding map with message length k and minimum distance d has block length $n \geq k + \frac{d-2}{2} \log_2(\frac{k}{d})$. You may use the fact that $\binom{n}{t} \geq \left(\frac{n}{t}\right)^t$ for any positive integers n and t with $n \geq t$.

4 Reed-Solomon Codes [9 points]

In this section, we shall now develop a family of codes for Alice and Bob to use, which uses small block length relative to message length.

It turns out that a $(k-1)$ degree polynomial can be uniquely determined from its evaluation at any k points. So, to allow for t erasures, we will transform a message m into a $(k-1)$ degree polynomial and encode it as the evaluation of that polynomial at $k+t$ different points. Then, if any t values in the codeword are lost, we can still recover m from the remaining k evaluation points.

We shall now formalize the above argument. Let q be prime and $q \geq n = k + t$

Now consider the following algorithm, given an input m

1. Let $m = m_1 \dots m_k$, and find $p(x) = m_1 + m_2 x + \cdots + m_k x^{k-1}$.
2. Compute $b_i = p(i) \mod q$ for $i = 1, \dots, n$.
3. Return $b_1 \dots b_n$.

The above is an encoding map $E_{RS} : \Sigma^k \to \Sigma^n$; let $\mathcal{C}_{RS}$ be the associated code. First, we shall find the minimum weight of $\mathcal{C}_{RS}$.

Problem 4.1 (Lagrange's Theorem, 2 points)

Let q be prime. Show that for any polynomial $p(x)$ with integer coefficients not all divisible by q, $p(x) \equiv 0 \mod q$ for at most $\deg(p)$ distinct values of x.

Problem 4.2 (2 points)

Show that $d(\mathcal{C}_{RS}) = n - k + 1$.

Using problem 1.5, we know that we can use a Reed Solomon Code to detect erasures and errors efficiently. But we still need to actually figure out an algorithm to do this.

Problem 4.3 (2 points)

Show that, for any $x_1, x_2, x_3, \dots, x_k$ and any y, we can find a polynomial f of degree $k-1$ with integer coefficients so that $f(x_1) \equiv y \mod q$ and $f(x_i) \equiv 0 \mod q$ for $2 \leq i \leq k$.

Problem 4.4 (3 points)

Devise an algorithm that allows you to, given the code $E_{RS}(m)$ of length $k+t$ with at most t erasures, decipher the original length k message m.

5 LT Codes [19 points]

Using the codes devised in sections 1 and 3, Alice and Bob have now devised an effective method of correcting for erasures and errors. Unfortunately, due to pollution, the water will now simply destroy $\frac{1}{2}$ of the messages Alice sends.

Alice wants to send Bob n messages, all of the same length. When Bob's received the n messages, he will send her a special passage in response. Unfortunately, Alice has no way to know whether her messages reach Bob, so they have to adopt a clever strategy to ensure the expected number of messages Alice must send is small.

Problem 5.1 (3 point)

One strategy would be for Alice to send all n messages one after another, then wait to see whether Bob has received them all, and send them all again if not. Bob will keep every message he receives until he has eventually received all of them. Show that, for sufficiently large n, the expected number of messages Alice sends is more than $10n$.

Suppose that Alice's messages are $m_1, \dots, m_n$. Suppose that Alice picks a random binary string s of length n, and then constructs a packet m_s by $\oplus$ing every message for which s has a 1. For example, if $n = 4$, $m_{1010} = m_1 \oplus m_3$. Each turn, Alice will generate a random s and send Bob the packet m_s.

Problem 5.2 (1 point)

Let e_i be the binary string of length n with a 1 at the $i$the entry and 0s elsewhere. Show that, if she can write $e_i = s_1 \oplus \cdots \oplus s_k$, then $m_i = m_{s_1} \oplus \cdots \oplus m_{s_k}$.

The span of a set of binary strings S is the set of strings which can be formed by $\oplus$ing some of them together. In other words, $\text{span}(S) = \{\bigoplus_{s \in T} s : T \subseteq S\}$. Let $\dim(S) = \log_2(|\text{span}(S)|)$

Problem 5.3 (3 points)

Show that, if $t \in \text{span}(S)$, then $\dim(S \cup \{t\}) = \dim(S)$; otherwise, if $t \notin \text{span}(S)$, then $\dim(S \cup \{t\}) = \dim(S) + 1$. Also show that, if S contains enough packets for Bob to decode all her original messages, then $\dim(S) = n$.

Suppose that Alice sends Bob the packets $m_{s_1}, m_{s_2}, \dots$. Let $d_i = \dim(\{s_1, \dots, s_i\})$, with $d_0 = 0$.

Problem 5.4 (10 points)

Alice begins sending packets to Bob.

(i) (5 points) Find the exact probability that Bob can recover all n original messages after n packets, and show that this probability is at least $\frac{1}{4}$ for all n.

(ii) (5 points) Show the probability that Bob can recover the n message after $n + 2$ packets is at least $\frac{3}{5}$ for all $n \geq 5$.

Problem 5.5 (2 points)

Show that the expected number of packets Alice must send until Bob has enough to decode all n messages is at most $8n$ for sufficiently large n. Remember that $\frac{1}{2}$ of the messages are lost.

6 Miscellaneous [12 points]

Now that Alice and Bob have figured out how to communicate, they are free to have fun working on random coding theory problems.

Problem 6.1 (4 points)

Consider a code with variable length $\mathcal{C} = \{1, 01, 001, 0001\}$ in which each of the codewords are transmitted with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$ respectively. What is the probability that a randomly chosen bit in a long stream of transmission is a 1?

Problem 6.2 (8 points)

Suppose that n people are dealt a red or black card, each independently with probability $\frac{1}{2}$. Each person holds up their card so that everyone can see it but themselves. Each person is given a chance to guess their own card, or to pass. The group wins as long as someone can guess which color card they have, and no one guesses wrong. Otherwise, they lose.

(i) (1 point) Give a strategy where the group wins with probability $\frac{1}{2}$.

(ii) (7 points) Suppose $n = 7$. Give a strategy where the group wins with probability $\frac{7}{8}$.