Math 328: Project 1 **Due Fri Feb 28**

The purpose of this project is a more in-depth investigation of the Hamming codes. You will discover an elegant decoding algorithm for the smallest non-trivial Hamming code and solve a well-known puzzle using the Hamming code. You will also learn a little bit about about finite fields and consider Hamming codes over arbitrary finite fields. Finally, there is an optional programming project if you are interested. You may work in pairs. Write (type) a paper that addresses the following questions.

---

1. **A Nice Decoding Algorithm for the Hamming Code**

   Recall that in the first day of class, we defined the binary $[7, 4, 3]$ Hamming code as
   $C_1 = \{(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \in \mathbb{Z}_2^7 : x_5 = x_1 + x_2 + x_4, x_6 = x_1 + x_3 + x_4, x_7 = x_2 + x_3 + x_4\}$

   The textbook gives a general description of binary Hamming codes through parity check matrices, and the $[7, 4, 3]$ Hamming code can be obtained by letting $r = 3$ in the general definition. Consider the following description of the Hamming code. Let C_2 be the code whose parity check matrix H (according to the standard definition of a parity check matrix) is the matrix whose columns are binary representations of integers 1-7. (For example, binary representation of 1 is [0 0 1], and of 6 is [1 1 0])

   (a) Show that the codes C_1 and C_2 are equivalent.

   (b) Use the parity check matrix of C_2 to devise a decoding algorithm that does not use cosets (or coset leaders) and correctly decodes received vectors when they contain exactly one error.

2. **Hamming Code and the Hat Puzzle**

   The binary Hamming code of length 7 has an interesting connection to a well-known puzzle: "The Hat Problem". The puzzle is as follows:

   At a mathematical show with 7 players, each player receives a hat that is red or blue. The color of each hat is determined by a coin toss. So the hat colors of players are determined randomly and independently. Each person can see the other players' hats but not his own. When the host signals, all players must simultaneously guess the color of their own hats or pass. The group shares a $1 million prize if at least one player guesses correctly and no players guess incorrectly.

   No communication of any sort is allowed between the players during the show, but they are given the rules in advance and they can have an initial strategy session before the game starts. What should they do to maximize their chances?

   (a) There is an obvious strategy with a 50% chance of winning. What is that?

   (b) Is there a strategy with a higher chance of winning?

   As you might predict, there is a strategy with a much higher chance of winning than 50%, and it makes use of the [7,4,3] Hamming code. The fact that the Hamming code is perfect and 1-error correcting makes this solution possible. In fact, that is the best possible solution for the puzzle. Describe how the Hamming code can be used to solve this puzzle with a much higher winning probability. What is that probability? Carefully explain and justify your algorithm. Why does it work?

3. **Hamming Codes Over Arbitrary Finite Fields**

   Although we mostly consider codes over the binary alphabet in this course (and it is the most important case), codes over other (larger) alphabets are also considered for both theoretical and practical purposes. Sometimes it is desirable to have an alphabet with more than just two symbols. Since a field is needed to consider linear codes (a field is needed to talk about a vector space), codes over arbitrary finite fields are considered in coding theory. For example, the on-line table of best known linear codes (www.codetables.de) contains the parameters of codes over the finite fields of order 2,3,4,5,7,8, and 9.

   In this part of the project, you are going to describe Hamming codes over arbitrary finite fields. You may not be very familiar with finite fields, but the following information about finite fields will be enough for this purpose.

   (a) For every prime power $q = p^n$, where p is a prime and n is a positive integer, there is a finite field with q elements denoted by $GF(q)$ or $\mathbb{F}_q$. (It is also referred to as "Galois field of order q"). For example, there is a finite field with 4 elements. Moreover, the field $GF(q)$ is unique up to isomorphism.

(b) The best known examples of finite fields are $\mathbb{Z}_p$, integers mod p for a prime p. Note that $\mathbb{Z}_4$ is not a field (so $GF(4) \neq \mathbb{Z}_4$).

(c) The finite field $GF(q)$ contains $\mathbb{Z}_p$ as a subfield (where q is a power of p, say $q = p^n$). We say that $GF(q)$ is an extension of $\mathbb{Z}_p$ (of degree n). For example, $GF(4)$ contains the binary field as a subfield. Give a description of the field $GF(4)$ by writing addition and multiplication tables. Let $GF(4) = \{0, 1, a, a+1\}$, where $a^2 = a + 1$.

Definition: Let r be an integer greater than 1. A q-ary Hamming code C is defined as the code whose parity check matrix H contains all vectors of length r over $GF(q)$ whose first non-zero component is 1 as its columns. (Note: This is according to the standard definition of the parity check matrix)

(a) What is the number of columns of H, i.e. what is the length of C?

(b) What is the dimension of C?

(c) Show that the minimum distance of C is 3. (Hint: Theorem 2.9.1 from the textbook also applies to codes over other alphabets but remember the difference in the definition of a parity check matrix)

(d) Obtain the general version of the sphere packing bound for $GF(q)$. (Give an argument to justify it) Then show that C is perfect (hence all Hamming codes are perfect)

(e) Implement the Hamming code over $\mathbb{Z}_5$, and $r = 2$ in Magma (find a parity check matrix and a generator matrix). Verify the code parameters. Encode the message $\vec{m} = (1, 0, 2, 4)$ into a codeword $\vec{c}$, then introduce an error at position 5 of $\vec{c}$ by adding a 4 to that position. Then, come up with a decoding algorithm for this (and general) Hamming code. Notice that there are two things to figure out in decoding: The position of the error, and the magnitude of the error.

4. **Bonus: Simulation Program for the Hat Puzzle**

Write a program (in a language or computer algebra system of your choice) to simulate the Hat Puzzle and its solution using the Hamming code. In every iteration, your program should generate a random distribution of hat colors, then determine players' response according to the best strategy obtained from the Hamming code. Run the simulation a large number of times and report the percentage of times the players win. Submit both the source code and the output. Report the result in a convenient, user-friendly format.