CS 70 Discrete Mathematics and Probability Theory
Spring 2016 Rao and Walrand Discussion 6B

1. **GCD of Polynomials**

   Let $A(x)$ and $B(x)$ be polynomials (with coefficients in $\mathbb{R}$ or $GF(m)$). We say that $\gcd(A(x),B(x)) = D(x)$ if $D(x)$ divides $A(x)$ and $B(x)$, and if every polynomial $C(x)$ that divides both $A(x)$ and $B(x)$ also divides $D(x)$. For example, $\gcd((x-1)(x+1),(x-1)(x+2)) = x-1$. Incidentally, $\gcd(A(x),B(x))$ is the highest degree polynomial that divides both $A(x)$ and $B(x)$.

   (a) Write a recursive program to compute $gcd(A(x),B(x))$. You may assume you already have a subroutine for dividing two polynomials.

   (b) Let $P(x) = x^4 - 1$ and $Q(x) = x^3 + x^2$ in standard form. Prove there are no polynomials $A(x)$ and $B(x)$ such that $A(x)P(x) + B(x)Q(x) = 1$ for all x.

   (c) Find polynomials $A(x)$ and $B(x)$ such that $A(x)P(x) + B(x)Q(x) = x+1$ for all x.

2. **(Berlekamp–Welch algorithm)**

   In this question we will go through an example of error-correcting codes with general errors. We will send a message (m_0, m_1, m_2) of length $n = 3$. We will use an error-correcting code for $k = 1$ general error, doing arithmetic modulo 5.

   (a) Suppose $(m_0, m_1, m_2) = (4,3,2)$. Use Lagrange interpolation to construct a polynomial $P(x)$ of degree 2 (remember all arithmetic is $\bmod 5$) so that $(P(0),P(1),P(2)) = (m_0,m_1,m_2)$. Then extend the message to lengeth $n+2k$ by appending $P(3),P(4)$. What is the polynomial $P(x)$ and what is the message $(c_0,c_1,c_2,c_3,c_4) = (P(0),P(1),P(2),P(3),P(4))$ that is sent?

   (b) Suppose the message is corrupted by changing c_0 to 0. We will locate the error using the Berlekamp–Welsh method. Let $E(x) = x + b_0$ be the error-locator polynomial, and $Q(x) = P(x)E(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ be a polynomial with unknown coefficients. Write down the system of linear equations (involving unknowns a_0,a_1,a_2,a_3,b_0) in the Berlekamp–Welsh method. You need not solve the equations.

   (c) The solution to the equations in part (b) is $b_0 = 0, a_0 = 0, a_1 = 4, a_2 = 4, a_3 = 0$. Show how the recipient can recover the original message (m_0,m_1,m_2).

3. **Error-correcting codes: An example**

   In this question we will go through an example of error-correcting codes with general errors. Since we will do this by hand, the message we will send is going to be short, consisting of $n = 3$ numbers, each modulo 5, and the number of errors will be $k = 1$.

   (a) First, construct the message. Let $a_0 = 4$, $a_1 = 3$, and $a_2 = 2$; use the polynomial interpolation formula to construct a polynomial $P(x)$ of degree 2 (remember that all arithmetic is mod 5) so that $P(0) = a_0$, $P(1) = a_1$, and $P(2) = a_2$; then extend the message to length $n+2k$ by adding $P(3)$ and $P(4)$. What is the polynomial $P(x)$ and what is the message that is sent?

(b) Suppose the message is corrupted by changing a_0 to 0. Use the Berlekamp-Welsh method to detect the location of the error and to reconstruct the original message $a_0 a_1 a_2$. Show clearly all your work.

4. **Counting Cantor**

Show that the Cantor set is uncountably infinite.

HINT: There are two standard ways to prove that something is uncountable: Find a bijection between it and some other uncountable set; or, use diagonalization.

Also, you might find it useful to know the following alternative definition of the Cantor set: S is the set of real numbers $x \in [0,1]$ that can be represented in base 3 (trinary) using only 0's and 2's (i.e., no 1's). (Be warned that there is some ambiguity in trinary representations: $1/3$ could be represented as either $0.10000\ldots$ or $0.02222\ldots$. For this definition, we require that the ambiguity be resolved by always using representations that end in $02222\ldots$ rather than $10000\ldots$, whenever you have a choice.)