Algorithms in the Real World (15-853), Fall 05
Assignment #2

Due November 2, 2005. Finish 100 points.

Problem 1: (20pt)

The ISBN is a 10-digit codeword such as 0-471-06259-6. The first digit indicates the language (0 for English), the next group specifies the publisher (471 for Wiley); the next group forms the book number assigned by the publisher. The final digit is chosen to make the entire number $x_1 \cdots x_{10}$ satisfy the single check equation: $\sum_{i=1}^{10}(ix_i) = 0 \pmod{11}$.
Note that the first 9 digits lie between 0 and 9, whereas the last digit can take any value between 0 and 10. The value 10 is represented by the letter X.

A. Calculate the check bit for the code 0-13-200809.

B. It is easy to see that the ISBN code can detect any single digit error. Show that the code can detect the transposition of any two digits (not necessarily consecutive).

C. The sixth digit in the code 0-13-28.796-X was smudged. Find the missing digit.

Problem 2: (20pt)

The NASA Mariner deep-space probes launched between 1969 and 1977 were equipped with a (32,6,16) Reed-Muller code for the transmission of pictures from the Moon and from Mars. As mentioned in class, Reed-Muller codes are the dual of extended Hamming codes. In particular the (32,6,16) Reed-Muller code is the dual of the (32,26,4) extended Hamming code. These codes have the feature that they can detect up to $n/2$ errors and are hence well suited for very noisy channels. Give three codewords for the (32,6,16) Reed-Muller code. These must be in systematic form—i.e. from a generator matrix in standard form.

Problem 3: (10pt)

Suppose that there is a very inexpensive PCI board that implements an $RS(255, 223)$ Reed Solomon encoder and decoder in hardware. (This is most likely true!) The board encodes or decodes sequences of 223 bytes and can correct up to 16 errors in a sequence. You would like to use Reed-Solomon codes to protect your data against errors as it is transmitted over a wireless communication channel. Unfortunately, your radio experiments show that, at your transmission rate, bursts of errors tend to be longer than 16 bytes. Using the $RS(255, 223)$-encoder/decoder as a building block, design a system that can correct up to 64 consecutive errors in a 1020-byte transmitted message, assuming that there are no other errors in the message. You must preserve the rate of the channel.

Problem 4: (50pt)

Please answer any four of the following questions. You may use Mathematica or any other tool to do polynomial arithmetic. You can even ask for help on using such tools, but please don't just take the results of using such tools from others.

A. Let α be a generator of GF(p^r). In general for polynomials GF$(p^r)[x]$ are $(x - \alpha), (x - \alpha^2), ...$ all the factors of $x^n - 1$?

B. How many binary cyclic codes of block length 15 are there? In particular, how many distinct generator polynomials are there?

C. Determine all codewords of a binary cyclic code of length 7 with parity check polynomial $h = x + 1$. Is this a useful code?

D. Let C_1 and C_2 be cyclic codes of block-length n generated by the polynomials $g_1(x)$ and $g_2(x)$ respectively. What is the generator polynomial for the smallest cyclic code containing $C_1 \cup C_2$? What is the generator polynomial for $C_1 \cap C_2$? (Note that the intersection of cyclic codes is cyclic).

E. Consider a $(3, 1, 3)_4$ RS code. The symbols are from the field GF(2^2) over polynomials modulo x^2+x+1 $(00 = 0, 01 = 1, 10 = x, 11 = x+1)$. Consider $\alpha = 10$ with $g = (x-\alpha)(x-\alpha^2)$. List the valid codewords (e.g. is 00 01 10 a codeword?). As in the notes and class, the leftmost symbol is the most significant. The answer might be trivial, but show your work.

Problem 5: (20pt)

A. Using the $r = 1/2$, $K = 3$ convolution coding scheme described in lecture, along with Viterbi decoding, show how the message 10011 would be encoded. Suppose that the 5th received bit is in error. Show how Viterbi decoding would correct the error.

B. For the same coding scheme (but not necessarily the same message), construct a not-completely-trivial example in which Viterbi decoding produces an incorrect answer. You should aim to minimize the number of errors used in the example, and your first two received bits must be correct.