Algorithms in the Real World (15-853), Spring 2018
Assignment #2 Due: 20 February 2018

Complete all problems.
You are not permitted to look at solutions of previous year assignments. You can work together in groups, but all solutions must be written up individually. If you get information from sources other than the course notes and slides, please cite the information, even if from Wikipedia or a textbook.
Please provide a simple illustration of your solution, more than just an answer.

Problem 1: A Non-Algorithmic Construction of Good Codes (10pt)

In lecture, we argued that for any point x on the n-dimensional hypercube, the number of hypercube points within distance D of x is $\mathsf{Vol}_n(D) := \sum_{i=0}^{D} \binom{n}{i}$. One can show that for $D \leq n/2$,

$$\log_2 \mathsf{Vol}_n(D) \quad \approx \quad n\, H(D/n),$$

where $H(p) := p \log_2(1/p) + (1-p)\log_2(1/(1-p))$ is the binary entropy function you've seen earlier in the course.

1. In other words, the Hamming lower bound says that the rate k/n of any binary code can be at most $\approx 1 - H(\lfloor \frac{d-1}{2} \rfloor / n)$. (Nothing to do here.)

2. Consider the following greedy algorithm to construct a distance-d code. Start with $\mathcal{C} = \emptyset$. Pick any vector from $\{0,1\}^n$ and add it to $\mathcal{C}$. Now, as long as there exists some vector that is at distance at least d from every codeword in $\mathcal{C}$, pick one of these and add it to $\mathcal{C}$. Show that the code $\mathcal{C}$ you construct has rate

$$\frac{k}{n} \geq 1 - \frac{\log_2 \mathsf{Vol}_n(d-1)}{n} \approx 1 - H(\tfrac{d-1}{n}).$$

Hence the lower and upper bounds differ in one being $\approx 1 - H(\frac{d-1}{2n})$ and the other $\approx 1 - H(\frac{d-1}{n})$. Closing this gap is a big part of coding-theory research.

Problem 2: Why are Reed-Solomon Codes Optimal? (10pt)

Another lower bound on the rate of codes is the following: for any $(n, k, d)_q$ code,

$$d \leq n - k + 1 \tag{1}$$

or equivalently, the rate of the code is at most $1 - \frac{d-1}{n}$. Note that Reed-Solomon codes satisfy this bound exactly, and hence are optimal (albeit only when q is large).

Prove this lower bound in (1). (Hint: what happens if you delete the first $d-1$ symbols of each codeword?)

Problem 3: Ever Wonder about ISBN? (15pt)

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, where the value 10 is represented by the letter X.

A. Give the parameters of this code in the $(n, k, d)_q$ notation. (I.e., what are n, k, d and q?)

B. Calculate the check digit for the message 0-13-201516.

C. 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).

D. The sixth digit in the code 0-621-5?157-2 was smudged. Find the missing digit.

Problem 4: Reed-Solomon Codes (10pt)

Suppose we have an inexpensive (and fast!) PCI board that implements an $RS(255, 223)$ Reed-Solomon encoder and decoder in hardware. (Assume you are working in the field $\mathbb{F}_{256} = \mathbb{F}_{2^8}$, and hence each symbol is one byte long.) The board encodes messages with $k = 223$ bytes, decodes received-words of $n = 255$ bytes, and corrects up to 16 byte errors in each codeword. 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 *bytes* of consecutive errors in a 1020-byte transmission, assuming that there are no other errors in the transmission . You must preserve the rate of the channel.

Problem 5: A Variant on LDPC codes (15pt)

Consider the following variant on LDPC codes. Like LDPC codes the code is given by a bipartite graph. But now, for each node on the right, the bits at its neighbors must form a proper Hamming code. To be concrete, each vertex on the right has degree 15, and the bits on its neighbors must form a $(15, 11, 3)$ Hamming code. Assume each vertex on the left has degree $d = 3$, so the number of nodes on the right is $n/5$.

1. What is the rate of this code (i.e. k/n)?
2. Assuming the bipartite graph has expansion (α, β) with $\beta = d/2 = 1.5$ prove that the code has distance at least αn. This is a similar argument to the one given in class for LDPC codes, but for LDPC codes we required that $\beta > d/2$.