Introduction to Information Theory, Fall 2019

Homework problem set #6 **due October 18, 2019**

> **Rules:** Always explain your solutions carefully. You can work in groups, but must write up your solutions alone. You must submit your solutions before the Friday exercise class (either in person or by email).

1. **Reed-Solomon codes (1 point):** Consider the Reed-Solomon code with the following parameters: $K = 2$, $N = 4$, $q = 7$, and $\alpha = 3$.

   (a) Encode a general message $s^K = [s_1, s_2]$ using the encoding algorithm from class.
   (b) Suppose that you receive $y^N = [0, 5, \perp, \perp]$, where $\perp$ indicates an *erasure error*. Decode the message using the decoding algorithm from class.

2. **Distance & Singleton bound (1 point):** Let $\mathcal{A} = \{0, \dots, q-1\}$. Given two strings x^N and $\tilde{x}^N$ in $\mathcal{A}^N$, define their *Hamming distance* $d(x^N, \tilde{x}^N)$ to be the number of places in which x^N and $\tilde{x}^N$ differ. Now suppose that $\mathcal{C} \subseteq \mathcal{A}^N$ is the set of codewords of an error-correcting code. We define the *distance* of the code to be

$$d := \min_{x^N \neq \tilde{x}^N \in \mathcal{C}} d(x^N, \tilde{x}^N).$$

   (a) Relate the distance to the number of *erasure errors* that can be corrected by the code.
   (b) Prove the following formula, known as the *Singleton bound*. It bounds the number of codewords in terms of the alphabet size, block size, and distance:

$$\#\mathcal{C} \leqslant q^{N-d+1}.$$

   Hint: Erase as many symbols as possible without changing the number of codewords.
   (c) Deduce that the Singleton bound is saturated for the Reed-Solomon codes from class.

3. **Reed-Solomon codes (1 point):**
   In this problem, you will implement the encoding and decoding algorithms for Reed-Solomon codes. To get started, open the notebook at https://colab.research.google.com/github/amsqi/iit19-homework/blob/master/06-homework.ipynb and follow the instructions.

   As always, please submit your solution as a Python notebook or script, or as a PDF printout. You can score the maximum score if your solution produces the correct output. We will only have a closer look at your code in case of problems.