# Homework 2, CS225 | ECE205A, UCSB, 2016 Winter

**Deadline, Copyright, and Academic Honesty:**

¶ *Your answers to this assignment should be uploaded as a pdf to the Gauchospace of the course no later than Thursday Week IV (2016-01-28, 23:59). The pdf should be produced using an appropriate typesetting program; it is not allowed to simply scan in your handwritten notes. Include your name and, if applicable, your perm. number on the pdf. Always explain your answer, but keep things succinct.*



¶ *You should complete these specific assignments by yourself. What you submit should be your own work; you are not allowed to copy or quote work by others. Students violating these or other rules of Academic Honesty will receive an "F" for the course and will be reported to the Dean of Students Office.*

**Question 1 (Kraft's inequality)**
Consider an alphabet of size $m$ and let $p_1, \ldots, p_m$ be a probability distribution over $\{1, \ldots, m\}$. Let $\ell_1, \ldots, \ell_m$ be code word lengths with $\sum_{i=1}^{m} 2^{-\ell_i} = 1 - \delta$ with $\delta > 0$ (i.e. this sum is strictly less than what Kraft's inequality demands). What can you prove about the difference $L(C) - H(p) = \sum_{i=1}^{m} p_i \ell_i + p_i \log p_i$? Try to work $\delta$ into your answer.

**Question 2 (Huffman Coding)**
Let $X \sim p(x)$ with $\mathcal{X} = \{a, b\}$, $p(a) = 0.25$, and $p(b) = 0.75$. Give the Huffman coding $C : \{aaa, aab, \ldots, bbb\} \to \{0, 1\}^*$ and the expected coding length for the random variable $X^3$.

**Question 3 (Shannon-Fano-Elias coding)**
Let $X \sim p(x)$ with $\mathcal{X} = \{a, b\}$, $p(a) = 0.25$, and $p(b) = 0.75$. For the random variable $X^3$ give the Shannon-Fano-Elias coding $C : \{aaa, aab, \ldots, bbb\} \to \{0, 1\}^*$ and the expected coding length per letter.

**Question 4 (Lempel-Ziv Compression)**
Consider the Lempel-Ziv coding algorithm on the repetitive binary string $01010101010101 \cdots$ The Lempel-Ziv algorithm codes the first 16 bits as follows: The 7 dictionary words of the first 16 bits are $0_1$, $1_2$, $01_3$, $010_4$, $10_5$, $101_6$, $0101_7$ and the LZ algorithm codes this as: $(0, 0), (0, 1), (1, 1), (3, 0), (2, 0), (5, 1), (4, 1)$. Using three bits for the indices this reads as the following 28 bit string: 0000000100110110010010111001.

Give an *upper bound* (as tight as possible) on the length of the LZ coding of the $n$ bit string $010101 \cdots$. You can phrase your answer using big O notation as $n$ grows (such that an upper bound of $n^2 + 7n - 3$ can be summarized as $O(n^2)$ and so on).