EEE 277 Machine Exercise #1: Huffman Coding

Use MATLAB/Octave/Python to do the tasks below. Your submission should include all the program files and generated output files. Submission should be a compressed file named 'EE277_ME1_<lastname>.zip'.

Overview:

Bits are the universal unit of information in today's digital world. Any information source can be represented using bits and bits can be transmitted reliably over an unreliable physical channels. In this lab, we study how to use bits to efficiently represent a source consisting of English text. Standard English texts are composed of English letters, numbers, and special symbols. For example, many texts contain only the 26 English letters (ignoring capital letters), 10 numerical digits, and 9 special symbols including " " (space), "," (comma), "." (period), "?" (question mark), "!" (exclamation point), " ' " (apostrophe), " " " (quotation mark), "-" (hyphen), and ":" (colon). This gives a total of 45 different symbols.

A simple way to represent these 45 different symbols using bits is to assign a unique binary string of length 6 to each symbol. Text-to-bit and bit-to-text conversions can be performed using a lookup table, which records the predetermined one-to-one correspondence between the text symbols and their binary representation. This simple approach uses the same number of bits to represent each of the text symbols. This may not be very smart considering that some text symbols appear a lot more often than the others. Assigning variable-length bit strings allows one to reduce the average length. This is the basic idea behind data compression.

Tasks:

1. [30 points] You are given a file named 'EE277_ME1_file2compress.txt". Estimate the probability of each symbols by counting their frequency of occurrence. Plot the result (using bar graph where y-axis is the probability and x-axis is the character). Save the plot in a .png file and/or .fig file. The name should be 'EE277_ME1_frequencyplot'
2. [30 points] Using your results in task 1, assign a binary codeword for each symbol using Huffman coding. Compute also the average codeword length. Create a document (a text file, doc file, or excel file) that contains the requested information. Name it 'EE277_ME1_codewordassignment'.
3. [20 points] Design an encoder that encodes the given text file. Save the encoder output to a textfile named 'EE277_ME1_encoder_out.txt'.
4. [20 points] Design a decoder that matches the encoder. The decoder will get input from 'EE277_ME1_encoder_out.txt' generated in task 3. Save the decoder output to a textfile named 'EE277_ME1_decoder_out.txt'