CSCI B505 – Fall 2018

Programming Assignment 5: Due online Nov 4 (SUN), 2018, 11:59pm EST.

Abstract: You will be implementing **Huffman code** (see page 431 in your textbook, and/or *07-Greedy-4* and *07-Greedy-5* in Module 8).

What to do:

1. Go to `http://www.gutenberg.org` and find a book of your choice, preferably English. Download the book as a plain text file via the link *Plain Text UTF-8*.
2. First, convert your original text so that it only uses 32 characters (i.e., the **lowercased** 26 English alphabet plus space, period, comma, exclamation point, question mark, apostrophe – discard everything else.) (If you want, you can choose to do 64 letters to support more symbols, or even do the full 128 letters of ASCII[1], but this is optional.)
3. Next, calculate the frequency of each letter/symbol in the text.
4. Now, create the Huffman Tree based on the character set from step 2 and the frequencies from step 3.
5. Make sure you can print out your code. For example:
   e: 000
   a: 010
   f: 111 ...

Discussion: Now you know how many bits are used by each alphabet/symbol in your code. Count the total number of bits required to encode your converted text from step 2, using your code (i.e., the size of the text that uses your encoding). Now count the number of bits required for the same text if you use a 5-bit fixed length code. (Note: If you used 64 characters, compare your code with a 6-bit fixed length code, and if you used 128, compare with a 7-bit fixed length code). Answer the following questions:

- **How many bits were you able to save by using Huffman encoding, compared to a n-bit fixed length code?**
- How many characters did you choose to go with – 32, 64, or 128?

What to submit:

- Your source code
- A PDF file (encouraged) for your discussion **AND** your printed Huffman code from step 5

Reminders:

- Ask questions on Piazza
- Keep an eye out for Piazza/Canvas announcements
- **Start early**

[1] `https://ascii.cl`