CSC 3102 Fall 2019
Programming Homework 2
due date 11/19/19, 11:59pm, 100 points

1. (60 points) You are given a dictionary of english words. Create a Trie, for storing these strings. To create the Trie, you can use repeated insertions. Compare two implementations of the Trie one where all the children of a given node are in a linked list vs use of hashing to jump to the correct child. In hash-based implementation, you must use the tuple pair (originating node id, next character) as a hashing key. In hash-table, you have to store a triplet (originating node id, next character, child node). For node-id, you can give every node a pre-order id after running pre-order on the constructed trie. When you compare the performance, you'll use 1000 word queries.

2. (40 points) Given a text file, find all the letter frequencies. You can sanitize the file by replacing every non-alphanumeric symbol into "white-space" character. Now, calculate frequency of each character. Create a Huffman Tree and Huffman code for each character. Write the sequential encoding of the full text file into a bit-array and then save this array into a file. Note: You don't need to pack bits into bytes - to simplify this implementation you can write it as a character for every bit. FInd the compression ratio you obtained. Also, write functions for decoding the huffman encoded file.

3. (Honors) Create Burrows Wheeler Transform (BWT) from the suffix array code given in the class. Now use Move-to-Front code to encode BWT. Compare the compression obtained with Huffman Compression above and write a report on this.