COP3530

Project 3 Commentary

Marco Austria

**Discuss the data structures you used. Why were they appropriate?**

The data structures used were Huffman Trees, Tree Maps, and Priority Queues.

The Huffman Trees were appropriate for organizing the frequencies of characters, so as to be able to encode characters, decode binary, and traverse a Huffman Tree easily. Tree Maps were used to store and order the characters and their frequencies based on ASCII. The Priority Queues were necessary in creating the Huffman Trees, which are built starting with the smallest nodes being combined.

**Discuss the computational complexity of the operations in the Huffman Algorithm, encoding and decoding, and traversing.**

Huffman Algorithm – O(nlogn) as it uses recursion on “n” characters in a greedy way using a priority queue with min being highest priority.

encodeFile() – O(nlogn) as it uses recursion through the tree on “n” characters.

decodeFile() – O(n) as it uses a while loop for every binary digit to travel through the tree.

traverseHuffmanTree() – O(nlogn) as it uses recursion for finding all “n” characters from a tree.

**Say something about what you learned.**

I learned a good deal of information in this project, a lot being related to the data structures implemented. I had never implemented any Huffman Trees, Tree Maps, nor Priority Queues before. I also got to learn a bit about Comparables because of the data structures. Lastly, I got to learn a lot more about recursion and the implementation in different situations in using it for traverseHuffmanTree() and encodeFile().