Andrew Jalbert

Professor Yau

CSCI 230 SP 17

30 April 2017

Project 3 Write-Up

Program Design Explanation: This program uses a priority queue to store the occurrence of characters that are read from an input. The lowest occurring character are stored at the front of the queue, and the highest occurring characters are found at the end of the queue. This implementation was chosen because it allowed easy removal of minimum valued characters, that in turn allowed easy creation of a binary tree. For my encodings, I chose to use a Hashtable. A hashtable allowed me to index each character’s encoding with a quick runtime. The character’s ascii value was the key, and stored at each index was a string representation of the code for each character. This allowed for quick access to each encoding given its ascii value. I chose to store the Huffman tree in my file as a preorder sequence. I made this choice because it seems easier to reconstruct the Huffman tree in this manner. Where a 1 indicates a leaf, and a 0 indicates an internal node. However, I could not build a proper tree from the stored preorder traversal. So, I excluded the tree from the compressed file. One cannot decompress the file unless it already has the tree made.

Project Experience: I found this project to be quite difficult. I encountered issues at every part of the project. The most troubling part was writing a compressed file, and then reading it for decompression. I could create the Huffman tree, generate codings for each character, and I could save my magic number and read the magic number from the file. After much trial and error, I was also able to save the coded tree and the coded message to the file. However, I could not construct a binary tree from the given preorder traversal, nor could I put in a proper EoF. This led to the binary string being translated in the file. I used a multitude of resources while attempting to solve this project. I used forum posts, videos, and classmates. I worked with Richard V. and Jameson D. They helped me analyze the algorithm, and implement it properly. I did enjoy this project even though I could not finish it. This project taught me how to write a file in bits instead of bytes. I also learned how complex some programming projects can be if one cannot break them down properly. Lastly, I learned how to compress and code files, which was very neat.