Tim Kraemer

Assignment 5 Design Document

In this assignment, we will be writing an encoder and a decoder, using Huffman Encoding, to be able to compress and decompress a file of bytes. Huffman Encoding works like this: a tree is created with nodes where the higher priority nodes (characters that appear more frequently in the file? are placed higher up in the tree. Nodes with a lower priority are placed farther from the top of the tree. A code is then assigned to each node, where a smaller code is used for higher priority nodes and a larger code is used for lower priority nodes. This is how a file gets compressed. The codes created by these nodes have less bytes compared to if you arranged the tree in a normal way.

Implementation:

The best thing to start off with is the I/O, since it is the start and end of the program. With this we will be able to read input files and write to output files.

an ADT for Nodes will be made, which creates the nodes for the huffman tree.

another ADT for the stack will be created, which focuses on a stack

an ADT for the priority queue will be made, which will help make the huffman tree.

huffman.c is a coding module that has functions that do different actions in the huffman encoding process

a Code ADT will be created, which specifies the function of the code creation for the encoding.

encode will be the implementation of the huffman encoder, and decode will be the implementation of the huffman decoder

Makefile will be made, but this makefile will be a bit different since we will have to have two targets, the encoder and the decoder.

The stack ADT will be similar to the stack adt we made in our last assignment, except we will include Nodes into our stack.