

Read Me:

Our program is able to compress either a file or directory, and can also decompress a file or directory. There are four flags b which builds a codebook and should be used when compressing a file or directory, c which can either compress a file or directory, and R which is used to compress a directory by recursively compressing the specified directory and all subdirectories. The d flag is used to decompress a file or directory, and the program will also need to be passed in the Huffman Codebook that was used to compress the file or directory.

The program runs at  $O(n \log n)$  efficiency, which is how long it takes to generate a Huffman tree for all the tokens and their frequencies, where  $n$  is the number of tokens. For every token a struct is generated and memory is allocated for it, for every unique token there is a node structure generated with memory allocated for it. This node structure is put into a minheap and then a Huffman tree, and there is also minheap struct that has memory allocated for it. All of these structures are freed before the program finishes running.