ECE 368
Project 2
Nausherwan Korai

Huffman Compression

We were given the task of creating a Huffman Compression algorithm. The basic idea is that we receive an input file, and need to compress it.

This is done by counting the frequency of the different ASCII letters in the file, and creating a codebook from these frequencies. What is this code book used for?

Well, the program assigns smaller bit values to represent often repeated letters and bigger bit values to less often recurring letters. This way, we are able to pack the same information into a lot less space using bit operations. The way the program does this is outlined below.

Compression

- Input file received
- Input file is read by frequency a function that creates a list of frequencies.
- A tree begins to be created from all of these nodes.
- The two lowest frequency tree nodes are joined together, and their frequencies are added together to make the parent node.
- This is continued until the tree is completed until the root.
- With this tree a code book is created, and each letter is assigned a number of bits, such as 01, 10, 110 etc.
- The header is then written to the file, and contains the binary tree that was used to create the codebook, new line is inserted.
- We traverse the file and start storing the letters in the file while referring to the codebook that was created using the tree.
- The rest of the file is written after the header.
- File is saved as filename + ".huff"

Decompression

- Input file is received
- Input file is read until \n is encountered. This is the header, which is the binary tree from which the codebook was created for the compression algorithm.
- The binary tree is reconstructed using the header.

- The rest of the file is now read, and as it is read, the binary tree is referred to and each value finds it's corresponding letter, and the file is decoded file by file until the end of the file is reached.
- File is saved as filename.huff + ".unhuff"

Format of file

The format of the compressed file is a binary file.

Huffman tree

A Huffman tree is used to create the codebook to compress the file and is used to build the tree to decompress the file. It is stored as the header in the compression file.

Code adoption

I adopted code from Professor Lu's book on C Programming. He was my professor during 264 and I found his material easy to read and understand. His algorithm also used a Huffman tree and it seemed prudent to pay attention to it. While professor Lu's code is separated into a lot of different files I chose to keep everything in one file.

Compression ratio

The compression ration is the size of the original file divided by the size of the compressed file.
From a test file of 175 bytes being compressed to 66 bytes, the compression ratio is: 2.65.

Notes

- I do not use any flags to compile the program.
- ./huff to compress
- ./unhuff to decompress
- 1 argument only: name of file.