

Text Compression and Decompression – Post-mortem

What went well

Compression

Overall the compression portion of the project went really well. Working out how the Huffman coding algorithm works via pen and paper and then implementing it in C++ was a good approach to solving each problem, and piece by piece I put together a compression programme that works. The compression mode of the programme reduces a file's size by approximately 40% in most cases, except when the original file is very small, as expected.

Decompression

Working out how the algorithm works was an educational experience, and via pen and paper I was able to work out an efficient process for reading the data stored in the binary file, and converting it back to ASCII text. Doing this also helped me work out what needed to be stored in the binary file during the compression process so that the data could be interpreted correctly.

Miscellaneous

This project also gave me the opportunity to learn how common UNIX-like command line utilities function, getting arguments, parsing them, and running different code paths based on what is passed to the programme. Since the purpose of this project was to create a potentially useful and usable compression utility, it also became apparent that a proper help/usage dialogue should be displayed if a user attempts to incorrectly use the programme. Implementing this was successful, and the relevant information is there, as it would appear in a typical man page.

What got in the way

Compression

This portion of the project took up more time than expected, and as a result, I was unable to complete the decompression portion of the programme within the allotted time. Several of the choices I made in the implementation significantly impacted the amount of time I spent on it, as I made them arbitrarily and without much thought. For instance, for storing the forest of trees used to build the Huffman tree, I had originally use a linked list, instead of a vector of nodes, which resulted in some painful debugging that took away from my productivity. As it turns out, this code was scrapped, and reimplemented using vectors instead.

Decompression

While this portion of the project should have been easier than the compression portion, it was not completed due to the scope of the compression algorithm.

Miscellaneous

I probably spent a little too much time working with pen and paper, and sorting out the usage of the programme, rather than working on the actual functionality.

What I could have improved

Compression

The execution time of the program is very long for some files of large size. The algorithm could probably be optimised at least a little bit, though the majority of the execution time is mostly file and string manipulation.

Decompression

This could have been implemented if I had the time to get around to it. I will probably continue to work on this through the summer, so that I have something that can be used as an example of my skills on my website (which I will also be hosting this summer).

Miscellaneous

The arguments processor could probably be cleaned up a little bit, but it would require more research on how command line parameters are usually interpreted.