

# Huffman Coding Assignment

vzbf32 | Z0973057

## Introduction

Huffman coding relies on assigning code words to characters, with the length of the code word being proportional to the frequency of the character in the encoding text. Each character is then replaced with a code word, and then the code word dictionary and the encoded text is written to a new file, which will have a smaller file size (on average).

I opted to implement canonical Huffman coding as it has a far smaller dictionary (and thus smaller encoded file size), although this does increase the CPU workload as the code words have to be reconstructed rather than simply read from the file.

## How to Execute the Code

### Encoder

To encode a file `some_text.txt`, navigate to the directory containing `huffman.py` and run:

```
python3 huffman.py -e some_text.txt
```

This will encode `some_text.txt` and store the output in `some_text.hc`.

### Decoder

To decode a file `some_text.hc`, navigate to the directory containing `huffman.py` and run:

```
python3 huffman.py -d some_text.hc
```

This will decode `some_text.hc` and store the output in `some_text_decoded.txt`.