

Compression Assignment – Philip Byrne 19449216

Task 3

<i>Huffman Algorithm</i>	mobydick.txt	genomeVirus.txt	medTale.txt	alphabet.txt
Number of bits	9531704	50008	45056	412
Number of bits after compression	5340944	12560	23752	656
Time to compress	0.09 seconds	0.006 seconds	0.006 seconds	0.006 seconds
Compression Ratio	0.56 : 1	0.25 : 1	0.53 : 1	1.59 : 1
Time to decompress	0.006 seconds	0.006 seconds	0.006 seconds	0.006 seconds
Number of bits after decompression	9531704	50008	45056	412

From these results we can see that both the compress and decompress methods work. We got the same number of bits after decompressing the compressed file as was in the original file. The compression worked well with all the files except for the case of alphabet.txt where it increased the size of the file. This is due to the alphabet.txt file consisting of one occurrence of each letter, lower case, and upper case. This caused the Huffman tree to be quite large so each 8-bit letter would become greater than 8 bits. With the other three files we saw that they at least halved in size and took almost no time, showing that the Huffman algorithm is efficient when dealing with files that contain few varied characters with different number of occurrences of each.

I tried compressing the already compressed genomeVirus.txt file and it went from being 12560 bits to 13968 bits, increasing in size with a compression ratio of 1.11. This occurs, perhaps because the file cannot be compressed any further using the Huffman algorithm as it only consists of 0s and 1s.

After compressing the bitmap file q32x48.bin with both my Huffman algorithm and the provided RunLength function, I found the Huffman algorithm to be more effective. The original file was 1536 bits. RunLength compressed this down to 1144 bits, while the Huffman algorithm compressed this down to 792 bits. They had a compression ratio of 0.74:1 and 0.52:1, respectively. This is probably due to the bitmap file containing few repeated values, making the Huffman algorithm more effective for this file.