Algorithms Assignment Report

Thomas Reilly - 18483722

Compression Test Results (Q1 & Q2)

File	Orignal SIze	Compressed Size	Decompressed Size	Compression Ratio	CompressionTime	Decompression Time	GCD (For calculating ratio)
mobydick.txt	1,191,463	667,651	1,191,463	1191463:667651	643687066	233176069	1
genomeVirus.txt	6,251	1,755	6,251	6251:1755	70122460	33204890	1
medTale.txt	5,632	2,987	5,632	5632:2987	49760811	18800182	1
algorithm.txt	8,039	4,574	8,039	8039:4574	68877499	42240219	1

Q3. What happens if you try to compress one of the already compressed files? Why do you think this occurs?

A null pointer exception occurs.

This happens because the compression file is made up of binary encoding which does not correspond to any characters. When the character frequencies are counted it doesnt find any and so, no nodes are created. (i.e.) the root of the tree is null. When the algorithm attempts to acess the root to create the lookup table it then produces a null pointer exception.

Q.4 Use the provided RunLength function to compress the bitmap file q32x48.bin. Do the same with your Huffman algorithm. Compare your results. What reason can you give for the difference in compression rates?

My Huffman Algorithm cannot be used to compress bitmaps as it requires ASCII text as input as mentioned in question 3.