Huffman Compression Analysis

Colm Rooney (19428806)

Step 1. Compress the provided text files.

Q1. Calculate the compression ratio achieved for each file. Also, report the time to compress and decompress each file.

Name	Uncompressed (bits)	Compressed (bits)	Time (Compress)	Ratio
mobydick.txt	9708968	5505432	0.1517111(s)	56.7%
medTale.txt	45872	24664	0.0528023(s)	53.8%
genomeVirus.txt	50008	12576	0.0366392(s)	25.1%
BeeMovieScript.txt	479624	280192	0.0501697(s)	58.4%
Q32x48.bin	1536	816	0.0323542(s)	53.1%

Step 2. Decompress the files you compressed.

Q2. Take the files you have just compressed and decompress them. Report the final bits of the decompressed files and the time taken to decompress each file.

Name	Compressed (bits)	Decompressed (bits)	Time (Decompress)
mobydick.txt	5505432	9708968	0.0640328(s)
medTale.txt	24664	45872	0.0079345(s)
genomeVirus.txt	12576	50008	0.01107(s)
BeeMovieScript.txt	280192	479624	0.0173705(s)
Q32x48.bin	816	1536	0.006167(s)

Step 3. Analysis of your results.

Assess the results of the above.

- From the data I can confidently assertain that the larger a file is, it becomes compressed at a higher ratio of compression.

Name	Uncompressed (bits)	Compressed (bits)	Time (Compress)	Time (Decompress)	Ratio
mobydick.txt	9708968	5505432	0.1517111(s)	0.0640328(s)	56.7%
medTale.txt	45872	24664	0.0528023(s)	0.0079345(s)	53.8%
genomeVirus.txt	50008	12576	0.0366392(s)	0.01107(s)	25.1%
BeeMovieScript.txt	479624	280192	0.0501697(s)	0.0173705(s)	58.4%
Q32x48.bin	1536	816	0.0323542(s)	0.006167(s)	53.1%

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

If an already compressed file gets compressed once again, I believe the compression ratio would be greater than 100%

Step 4. Run length.

Q4. 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?