Algorithms Assignment 2

Dawid Skraba 19433692

Task 3.

Q1.

File	Original Size	Compressed Size	Time to compress(ns)	Time to decompress(ns)	Compression Ratio
Random.txt	2Kb	881b	47,223,857	10,101,733	2:1
genomeVirus.txt	6Kb	2Kb	42,003,196	9,046,989	3:1
MedTale.txt	6Kb	3Kb	37,365,925	13,273,788	2:1
mobyDick.txt	1.2Mb	668Kb	193,431,384	76,316,045	2:1
q32x48.bin	192b	102b	40,603,929	3,332,927	2:1

Random.txt decreased by 56%
genomeVirus.txt decreased by 67%
MedTale.txt decreased by 50%
mobyDick.txt decreased by 45%
q32x48.bin decreased by 47%

Most files using Huffman compression decreased in size around 50%. The only file that considerably decreased more was the genomeVirus. This file only contains four distinct characters making it a lot easier for the Huffman algorithm to produce a small trie of the file. Times to compress depended on the size of the original file. The times to decompress a file we significantly shorter. This happened because it takes less time to search using values rather than keys when traversing the trie.

Q2.

File	Decompressed Size	Time to decompress
Random.txt	2Kb	10,101,733
genomeVirus.txt	6Kb	9,046,989
MedTale.txt	6Kb	13,273,788
mobyDick.txt	1.2Mb	76,316,045
q32x48.bin	192b	3,332,927

Q3.

After compressing a file that has already been compressed it grows larger. This is the case because data cannot be compressed any further using Huffman encoding if no data were to be lost (lossless). So, the result of compression remains the same size, but it has an encoded trie added to the start. This is what makes the file bigger in size.

Q4.

Original file size of q32x48.bin is 192 bytes. After using Huffman compression, the file size shrinks to 102 bytes. After using RunLength compression the file size shrinks to 143 bytes.
The compression rate for Huffman is: 1.9:1
The compression rate of RunLength is: 1.3:1
RunLength was 21% less effective than Huffman compression. This may be because the file we are compressing has a small number of distinct bytes. This would mean that Huffman

encoding would be better at encoding this file than RunLength. RunLength is good at compressing files which have long continuous characters and apparently this file isn’t structured in this way.