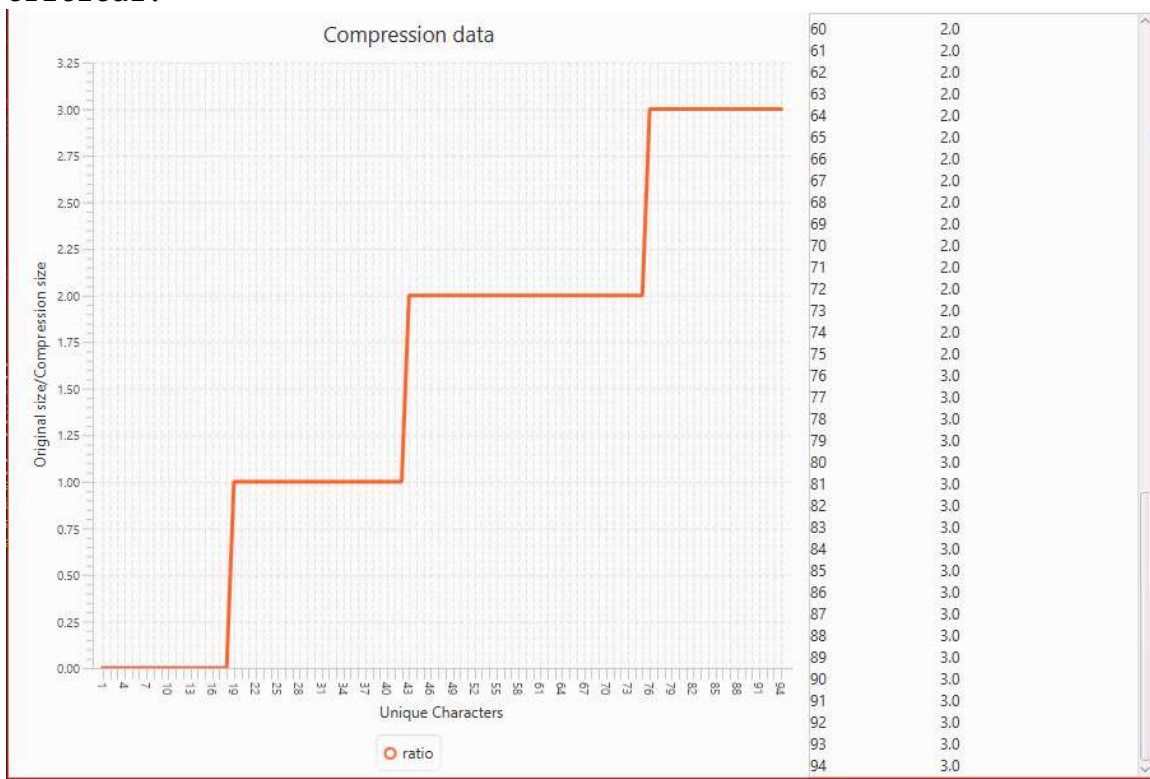
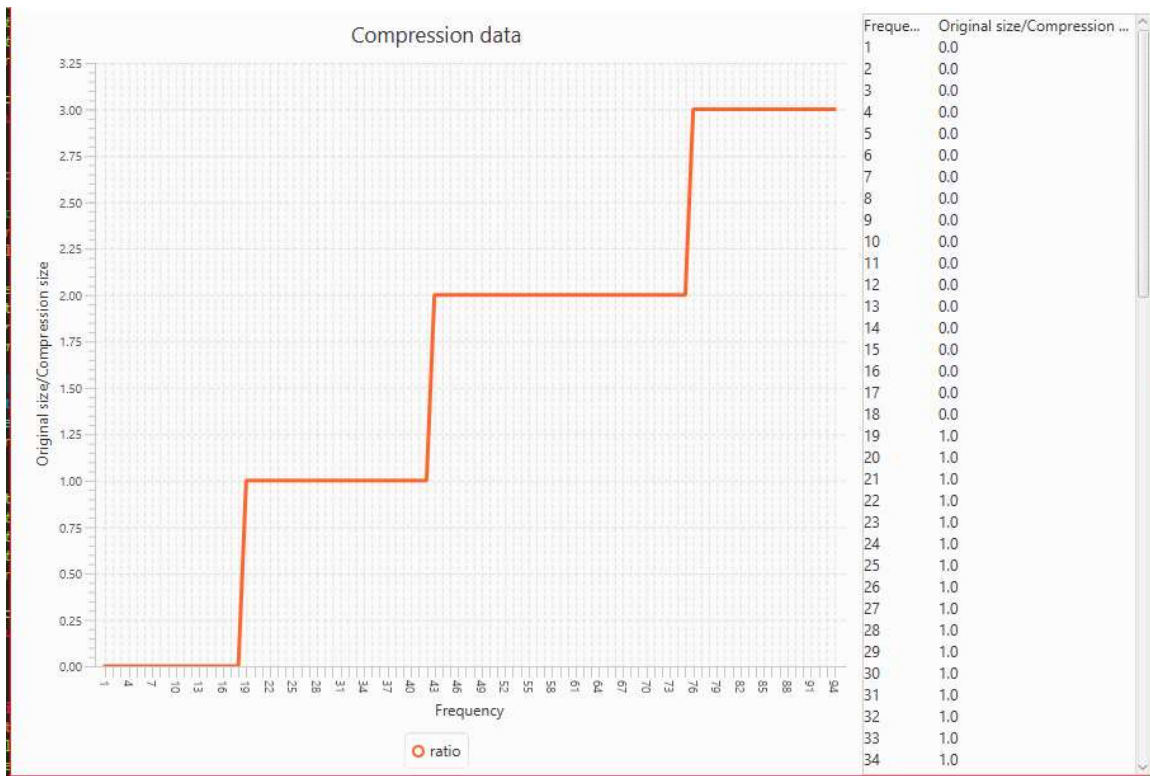


Nickolas Komarnitsky
Analysis12

1. Design and conduct an experiment to evaluate the effectiveness of Huffman's algorithm. How is the compression ratio (compressed size / uncompressed size) affected by the number of unique characters in the original file and the frequency of the characters? Carefully describe your experiment, so that anyone reading this document could replicate your results. Submit any code required to conduct your experiment with the rest of your program and make sure that the code is well-commented. Plot the results of your experiment. Since the organization of your plot(s) is not specified here, the labels and titles of your plots(s), as well as, your interpretation of the plots is critical.



For this experiment I created 94 files, naming them 0.txt, 1.txt, etc. The name told how many unique characters were in the file, with a maximum of 93 unique characters being possible. I then took the size of the original file\ the compressed file and then graphed it based on the number of unique characters that were in that file. The compression ratio increased as more unique characters were in the file.



In this experiment I created 94 files in which I added unique characters to a file with increasing frequency of each character. For example, the 3rd file would be `_ _ _ !!! ""` (lines represent spaces). The graph above shows that as the frequency of the characters increases the ratio increases.

2. For what input files will using Huffman's algorithm result in a significantly reduced number of bits in the compressed file? For what input files can you expect little or no savings?

I can expect that in a file with only one instance of each letter, there will be little to no savings. In a file with multiple of every letter there will be a significantly reduced size.

3. Why does Huffman's algorithm repeatedly merge the two smallest-weight trees, rather than the two largest-weight trees?

In order to ensure that the least occurring elements are using the most bits.

4. Does Huffman's algorithm perform lossless or lossy data compression? Explain your answer. (A quick google search can define the difference between lossless and lossy compression).

Lossless, the decompressed file is the exact same as the original file, it doesn't lose any of the letters in the file or where they are.

5. How many hours did you spend on this assignment?
10-15