Rebekah Peterson
CS 2420
Assignment 12: Huffman Compression

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?

Experiment: To evaluate the effectiveness of Huffman's algorithm, this experiment explores the compression ratio of files based on uniqueness or frequency of characters. Unique files consisted of a greater variety of characters, while frequency files had multiple duplicates.

Explanation: We can see from the graph that files that have many unique characters don't compress as well as those that have characters that appear more frequently and practically unused characters.

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?

The smaller the portion of all possible characters that are used, the greater potential for reduced number of bits. Input files that only use a portion of all possible characters will be reduced the most. This algorithm wouldn't be useful for files that use all (or close to all) of the possible characters.

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

By merging the two smallest-weight trees, characters that appear less often in the file will have longer bit codes, while characters that appear more often will have the shorter bit codes. This reduces the overall size of the file more than if the algorithm merged the two largest-weight trees..

4. Does Huffman's algorithm perform lossless or lossy data compression? Explain your answer.

Huffman's algorithm performs a lossless data compression. This means that all of the original data is recoverable upon decompressing..

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

4 - 6 hours