CSE310 Fall2020

PROJECT 1 REPORT

Name: Sunjeet Jena
ASUID: 1218420294

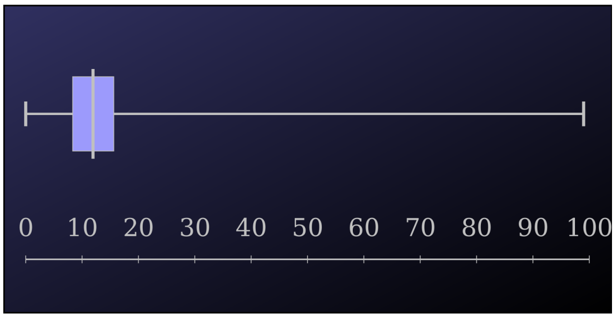
1) The experiments were ran on the sample input provided with the project description i.e Small Sample Input, Medium Sample Input and Large Sample Input.

Small Sample Input files has less than 5 lines in each file and each line has around 0-20 characters.
Medium Sample Input files has 150-750 lines in each file and each line has around 0-100 characters.
Large Sample Input files has 4000-16000 lines in each file and each line has around 0-100 characters.

Multiple experiments were ran against the inputs as mentioned above. First experiment was ran to observe the average compression ratio, its standard deviation, minimum and maximum values.
Second experiment involved observing the encoding time with Insertion Sort and Quicksort.
In the third experiment, decoding time was observed for both Insetion Sort and Quicksort.
And in the fourth experiment, compression ratio as a function of number of lines encoded was observed.

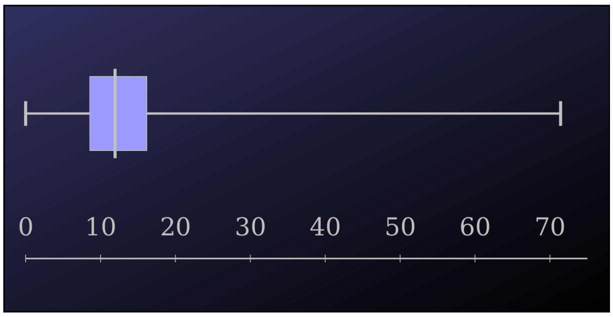
2) Average Compression Ratio in each sample Input and their corresponding Box Plots:

Large Sample Input:



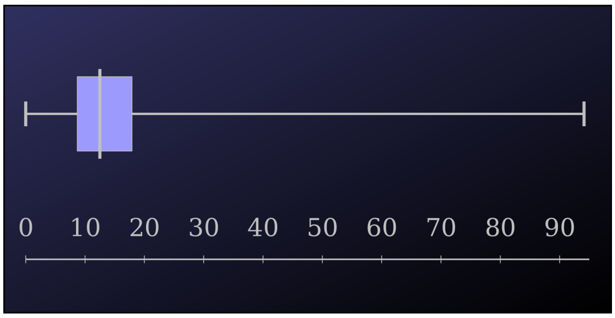
Filename: anne-of-avonlea.txt

Average: 12.867
Minimum: 0
Maximum: 99
Standard Deviation: 8.37833



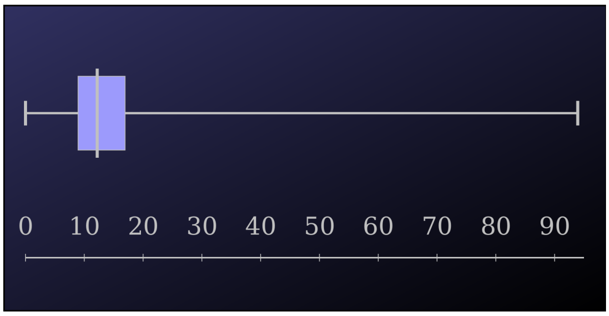
Filename: tale-of-two-cities.txt

Average: 12.617
Standard Deviation: 6.8064
Minimum: 0.0
Maximum: 71.4286

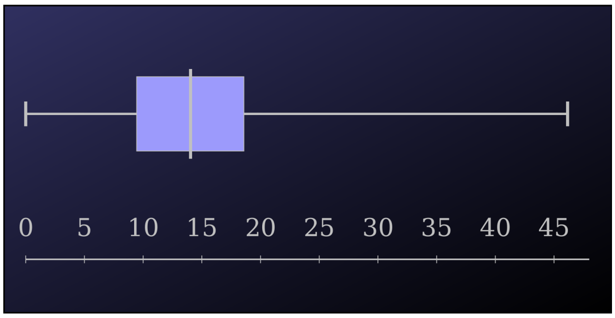


Filename: through-the-looking-glass.txt
Average: 14.8847
Standard Deviation: 10.9551
Minimum: 0.0
Maximum: 94.1176

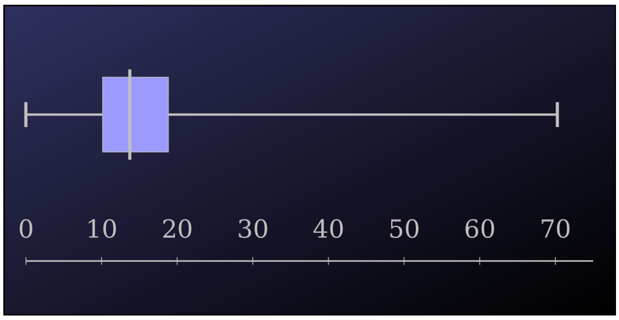
Medium Sample Input:



Filename: anne-of-avonlea ch1-and-2.txt
Average:14.3361
Median: 12.1951
Minimum: 0
Maximum: 93.9394
Standard Deviation:10.915120



Filename: tale-of-two-cities ch1.txt
Average:14.3361
Median: 12.1951
Minimum: 0
Maximum: 93.9394
Standard Deviation:10.915120



Filename: through-the-looking-glass-ch1.txt
Average:16.2792
Median: 13.7413
Minimum: 0
Maximum: 70.2703
Standard Deviation: 10.7212

Small Sample Input: (*Note: Box plot for these samples are not possible as the data is less than 5)

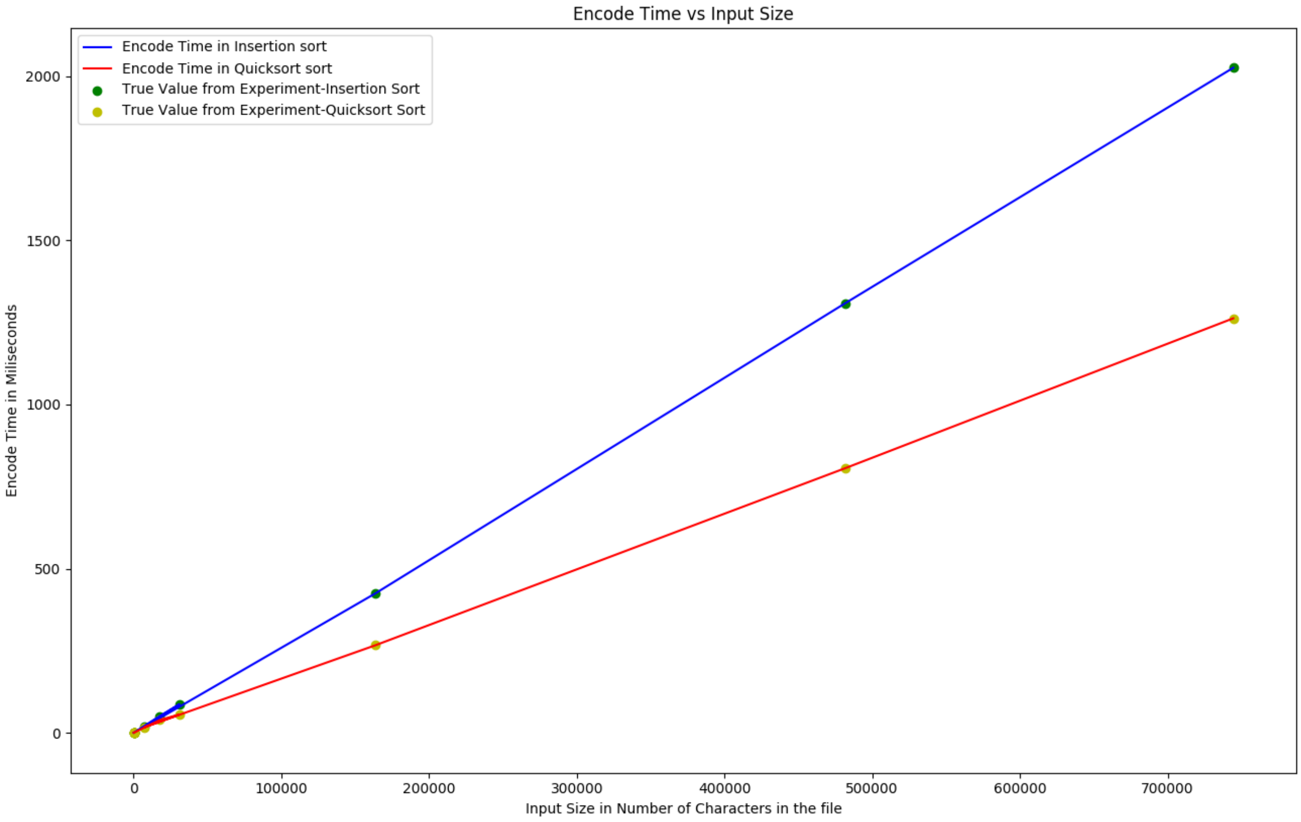
Filename: haiku1.txt
Average: 8.0409
Standard Deviation: 6.4493
Minimum: 0.0
Maximum: 15.7895

Filename: haiku2.txt
Average: 8.09523
Standard Deviation: 2.3650
Minimum: 4.7619
Maximum: 10.0

Filename: haiku3.txt
Average: 6.428
Standard Deviation: 5.9189
Minimum: 0.0
Maximum: 14.2857

Filename: haiku4.txt
Average: 3.6111133333333334
Standard Deviation: 2.749
Minimum: 0.0
Maximum: 6.66667

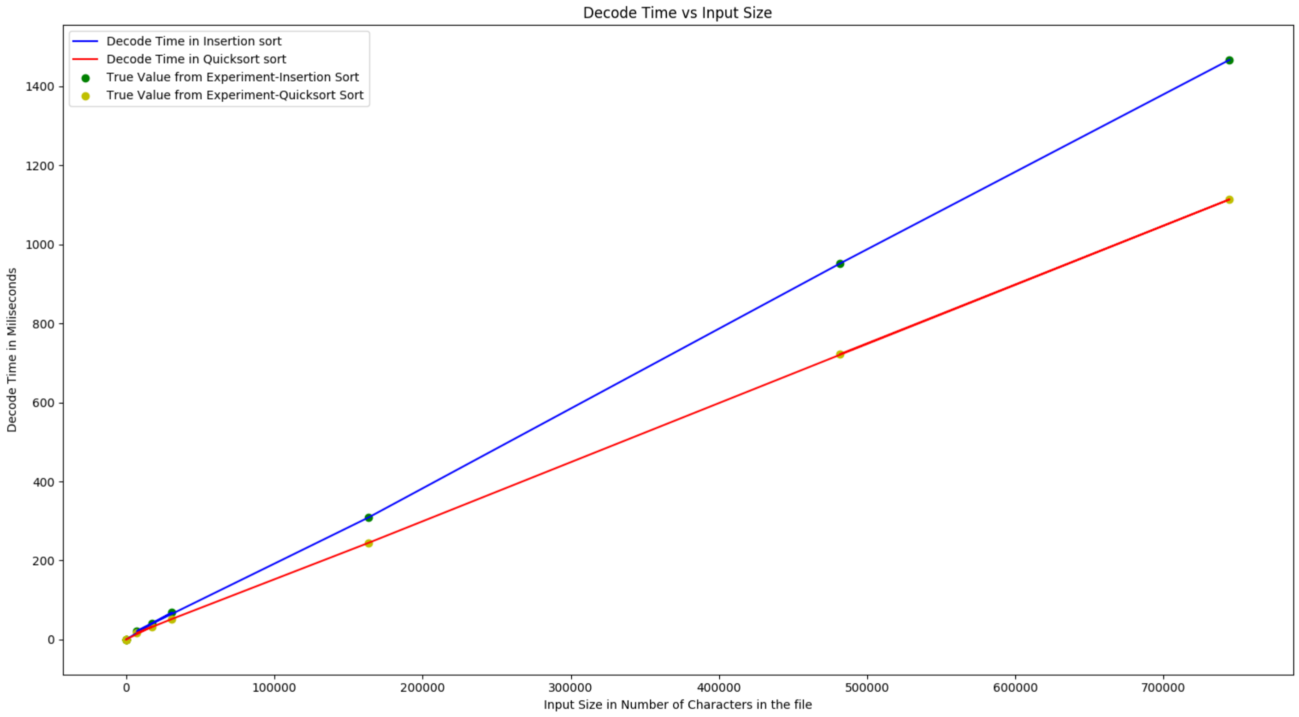
3) Time to encode each input for each sort, i.e. for Insertion and for Quicksort



The point in the graph represent the true time in miliseconds. The x-axis represents the number of characters in the file and y-axis represent the time taken for the encoding. Blue line follows the time taken with insertion sort and red line follows the time taken with quicksort.

We can clearly see that as the input size increases, the performance difference between Insertion Sort and Quicksort increases. Smaller file size lie to the left of the graph and Larger file size lie to the left of the graph. The Encoding is linear with the size of the input file.

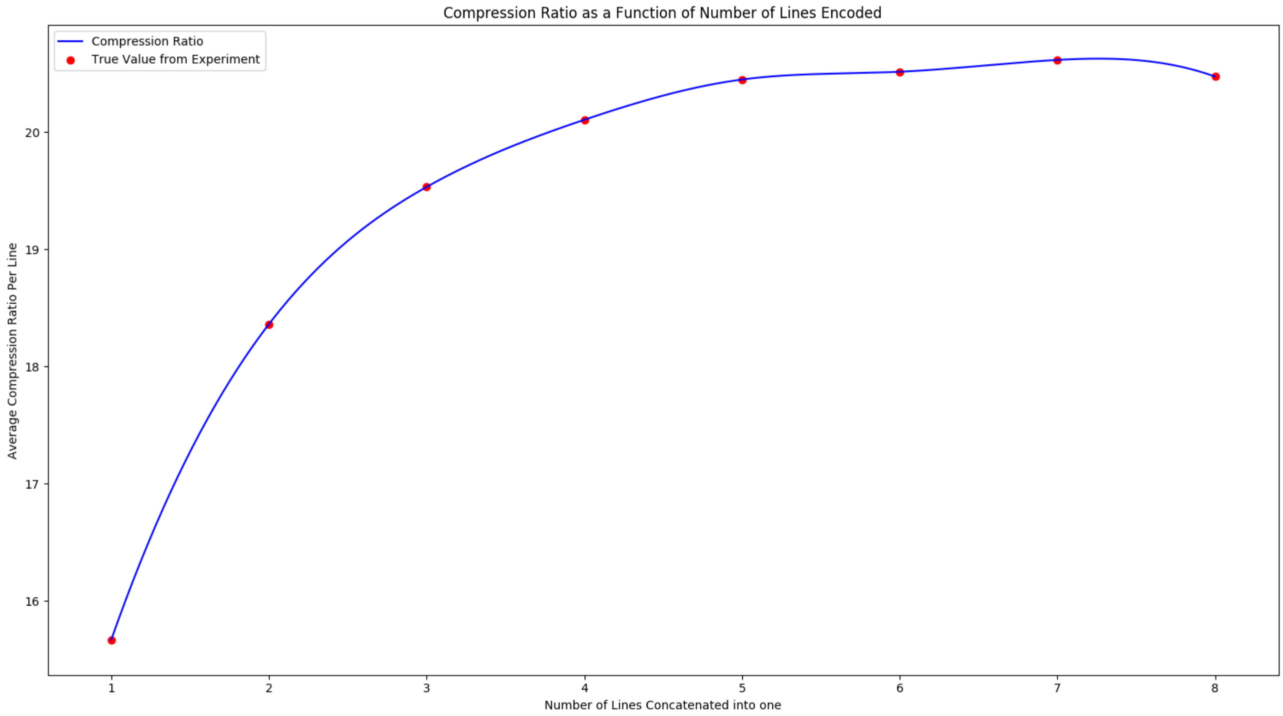
4) Time to decode each encoded input.



The points in the graph represent the true time in miliseconds. The x-axis represents the number of characters in the file and y-axis represent the time taken for the decoding. Blue line follows the time taken with insertion sort and red line follows the time taken with quicksort.

We can clearly see that as the input size increases, the performance difference between Insertion Sort and Quicksort increases. Smaller file size lie to the left of the graph and Larger file size lie to the left of the graph. The decoding time is linear with the size of the input file.

5) The compression ratio as a function of number of lines encoded.



The red points in the graph represent the true average compression ratio per line. The x-axis represents the number of lines encoded(number of line concatenated before encoding) and y-axis represents the average compression ratio per line over a file. Blue line follows the average compression ratio per line.

The graph has been drawn with the inpur taken from Large Sample Input “through-the-looking-glass.txt”, which was provided as the sample test input.

In the graph we can clearly see that as the the number of line that are concatenated into one before encoding increases, so does the compression ratio. But after sometime, it hits a peak and after that the average compression ratio doesn’t change much with the increase in number of lines.