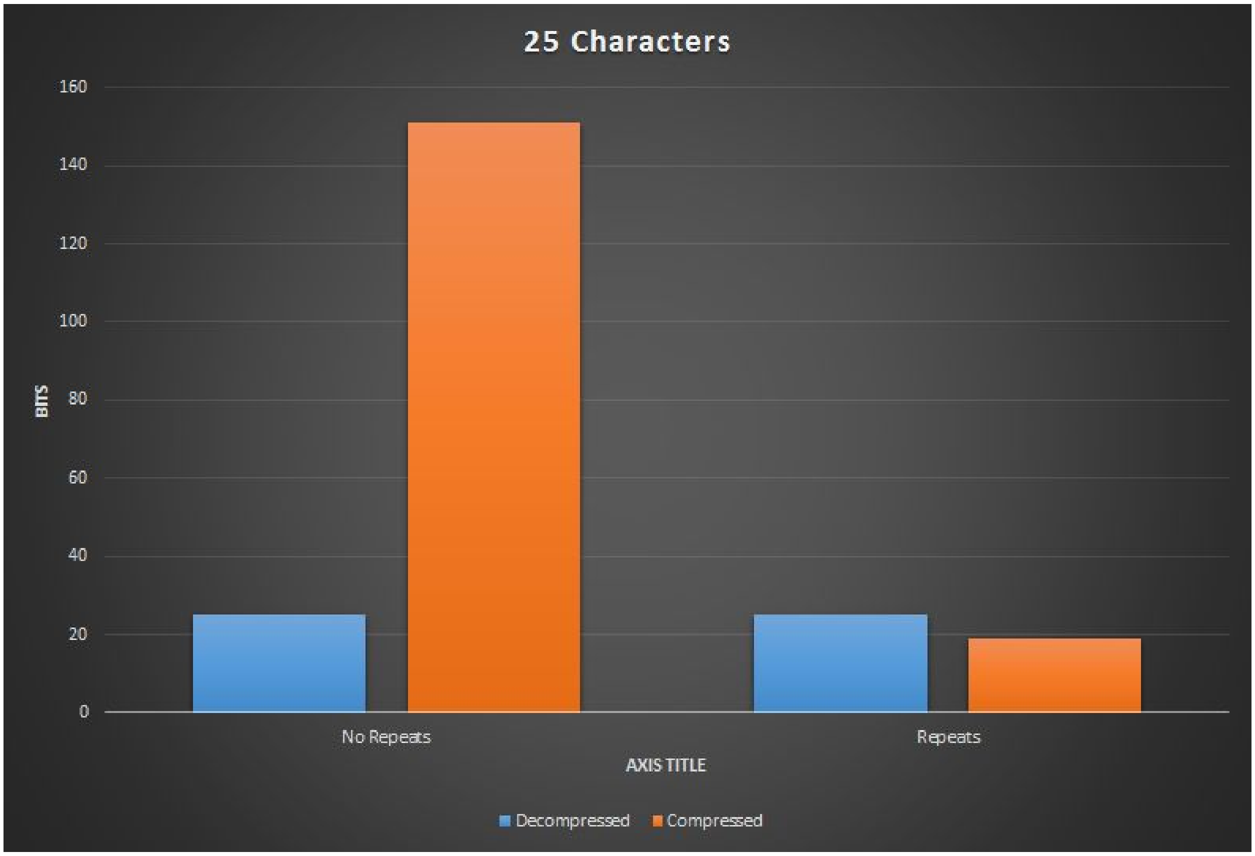
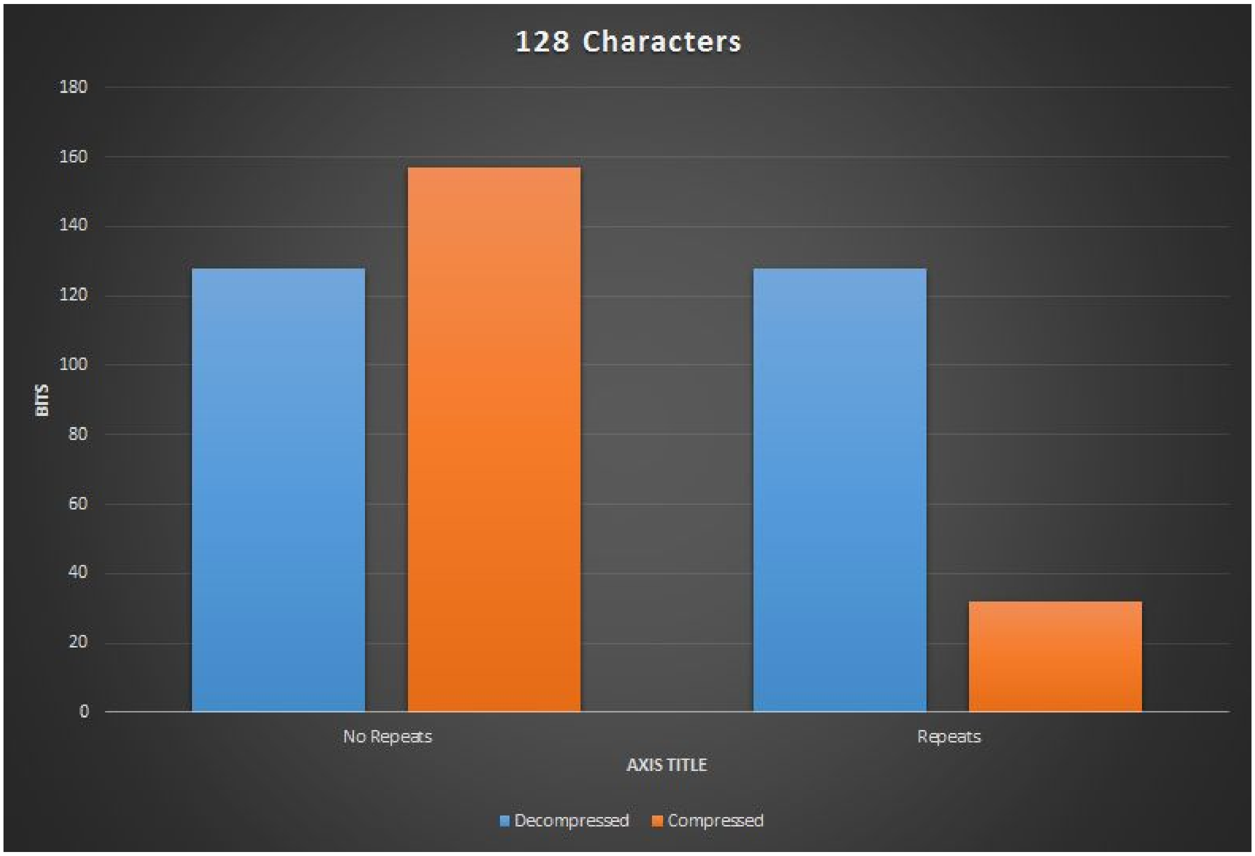
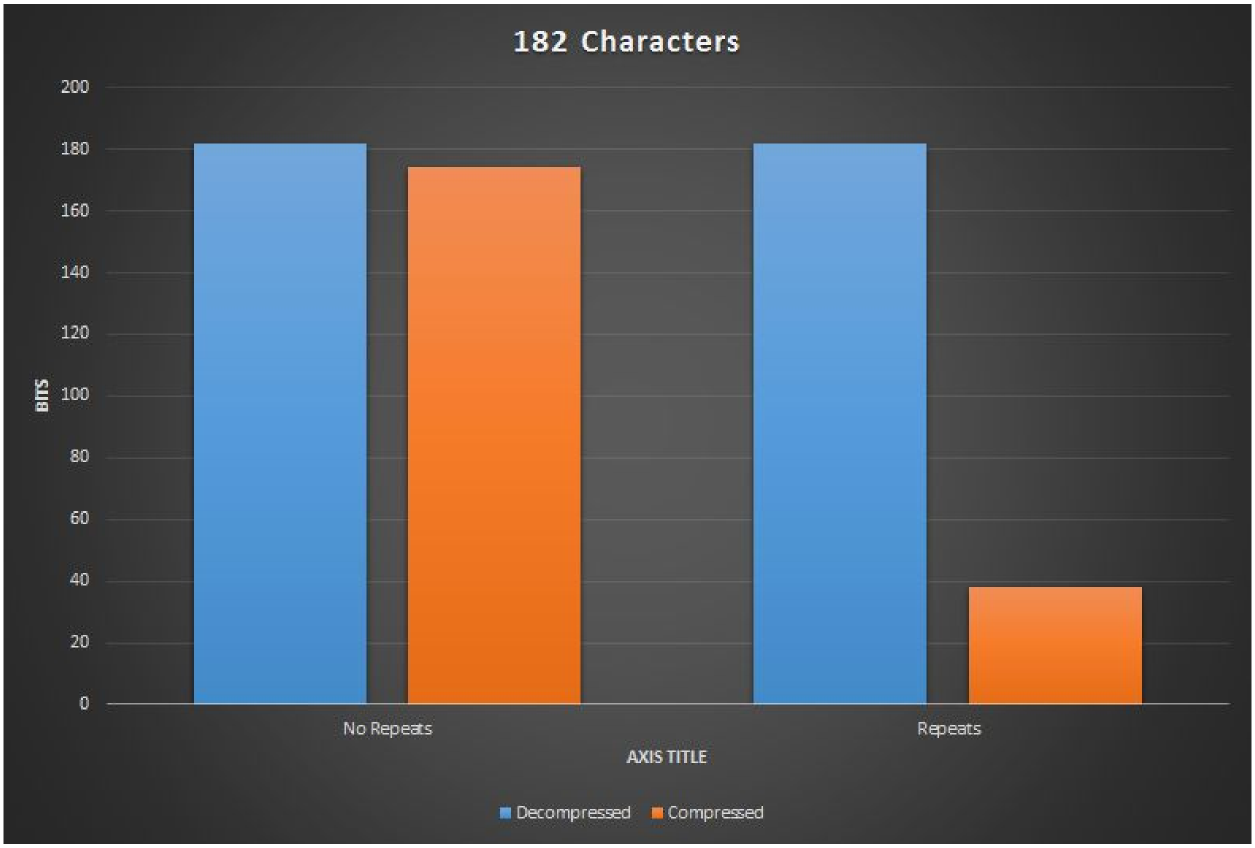
Cooper Pender (u0843147)
11/18/2016
Assignment 12 Analysis

1. For my experiment, I wrote several text files with repeated characters, and several file with no repeated characters. After I did this I took the original uncompressed size and then compressed each file using Huffmans compression. When that was done I computed the compression ratio. I just used the compressionDemo class provided to us to conduct this experiment.



This data shows that with smaller files, and files with non repeated characters compression is not nearly as useful as if we have a larger files, with some amounts of repeated characters, which is what most files fall under.


128 Characters
180
160
140
120
100
80
60
40
20
0
BITS
No Repeats
Repeats
AXIS TITLE
■ Decompressed
■ Compressed


182 Characters
200
180
160
140
120
100
80
60
40
20
0
BITS
No Repeats
Repeats
AXIS TITLE
■ Decompressed
■ Compressed

2. Huffmans algorithm will perform significantly better and compress much more when there are duplicate characters. This is because Huffmans uses weights, and can represent a repeated character easily. This algorithm will show little to no benefit from a file with no repeated characters.

3. Huffmans takes out the characters with the lowest weight first and merges those, since the less frequent a characters is in the original file, the lower on the tree we want it to be.

4. Huffmans algorithm performs a lossless encoding of data. It is defined as a lossless compression because through the process of encoding and decoding, there is no original data that is lost, or deleted.

5. Around 5 hours.