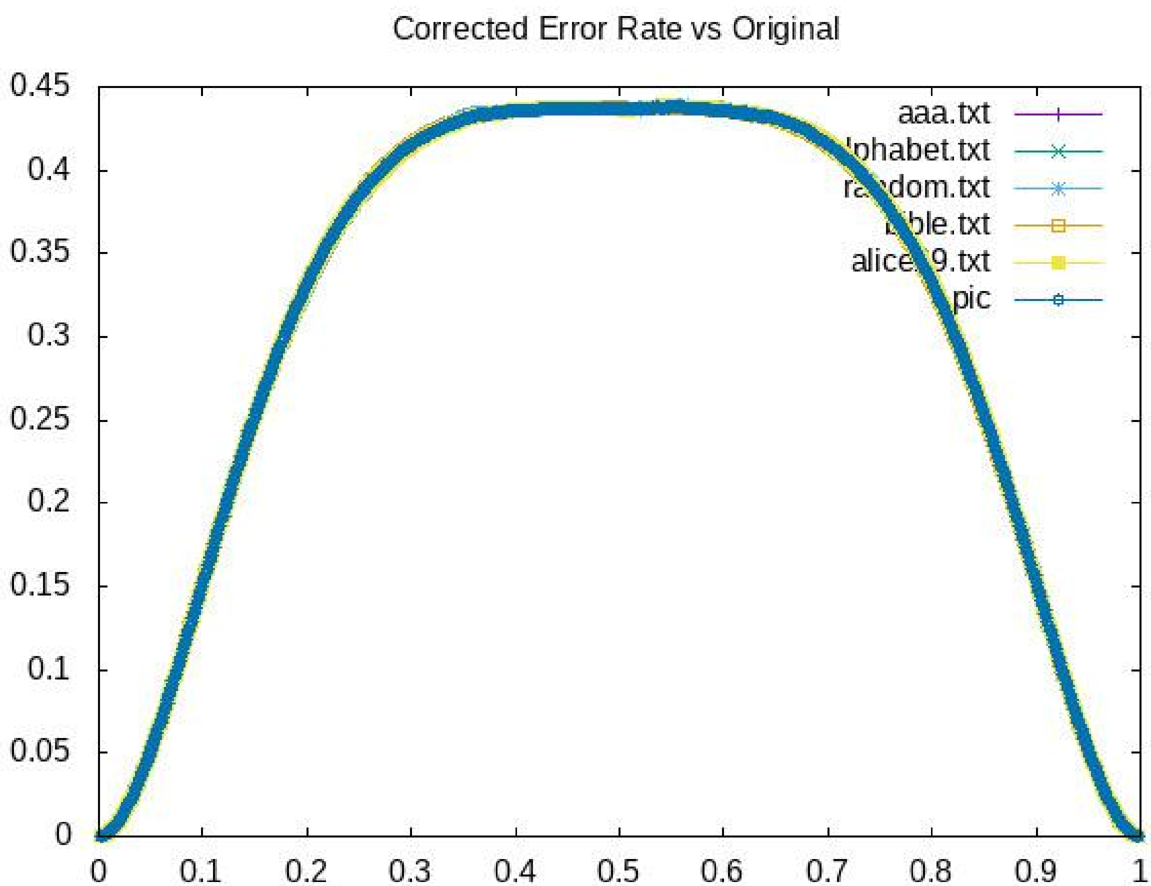
Bill Zhang
jzhan411@ucsc.edu
5/10/2021

CSE13S Spring 2021
Assignment 5: Hamming Codes
WriteUp

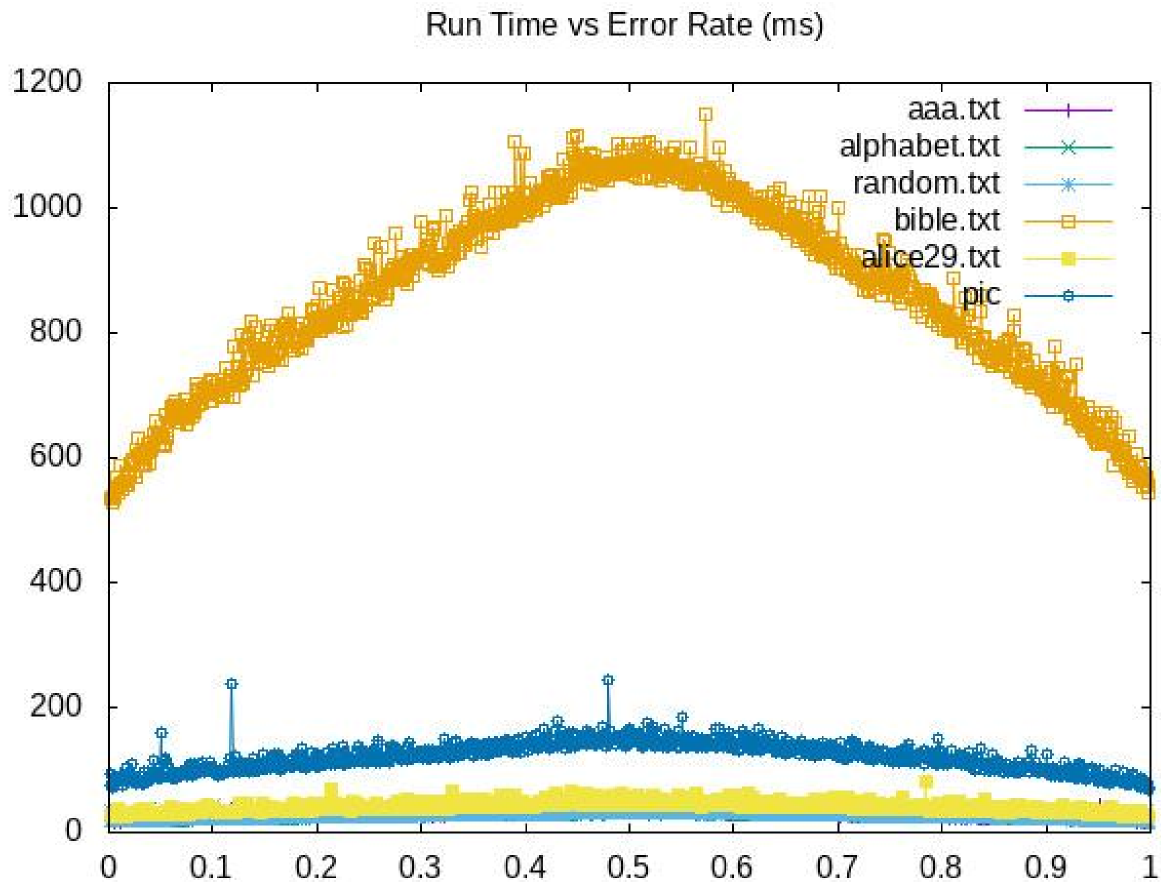
**Correcting Errors:**

This is the graph of the rate of uncorrected errors vs the error rate applied to the hamming code. **The title of the graph is incorrect, supposed to be uncorrected error vs error rate.**



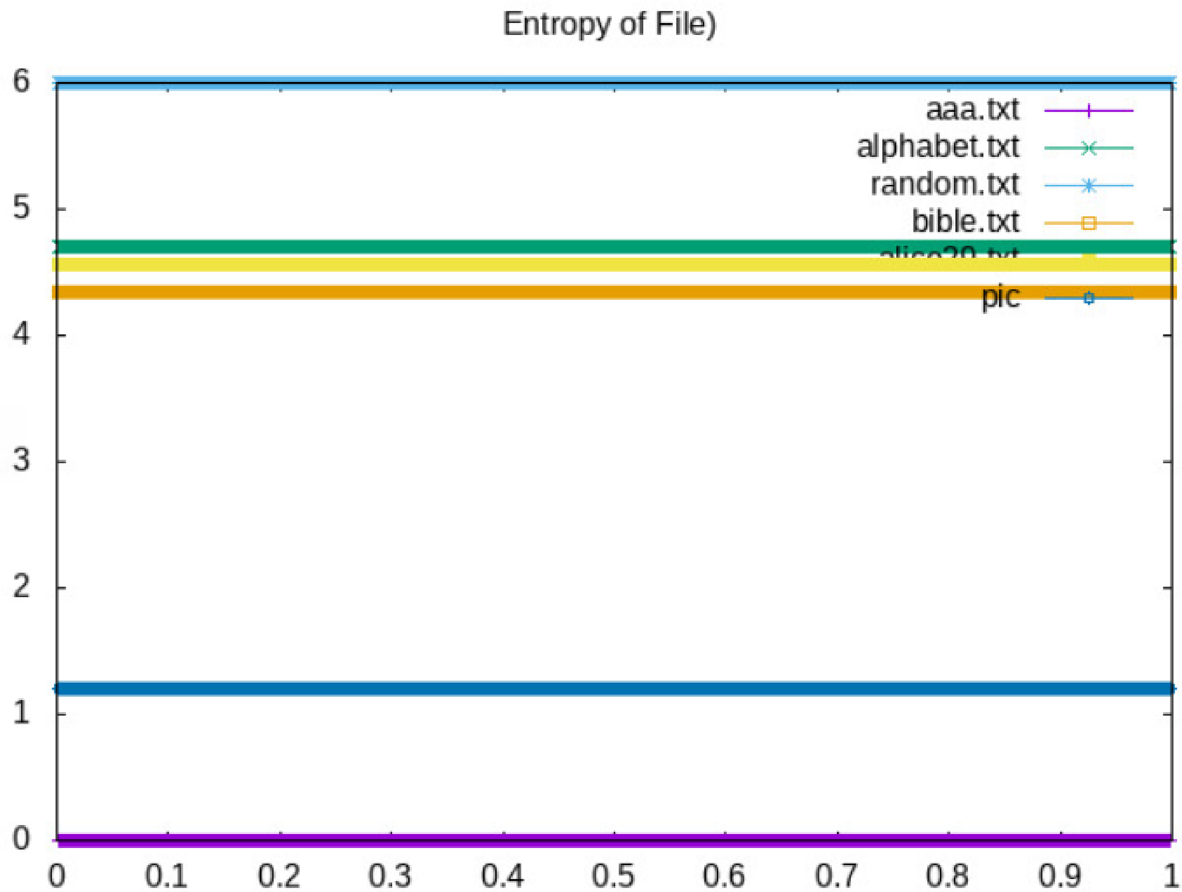

In the beginning and end, the uncorrected error rate is low. For the beginning, there are not a lot of errors, thus not a lot of errors that are uncorrectable. At the end, there are way too many errors in the Hamming code, thus the incorrect hamming codes start looking like valid hamming codes, and the decoder will start incorrectly decoding them. Thus the uncorrected rate drops again. At around 0.2 to 0.8 error rate, the uncorrected rate is around 25%. The maximum amount of uncorrected errors is slightly under 45%, from 0.3 to 0.7 error rate.

**Run Time:**



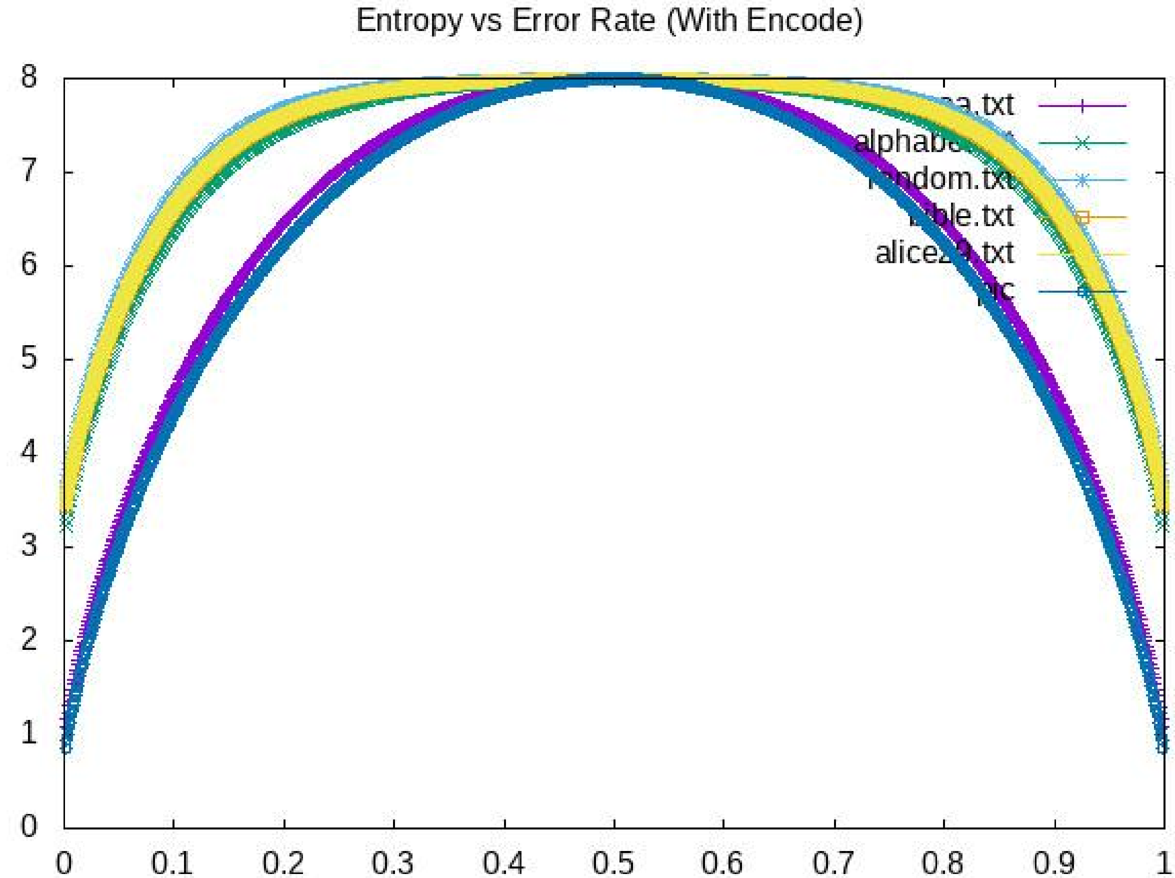

This is the graph for run time it takes to encode then decode a document. The Run time for smaller graphs is miniscule compared to larger graphs like bible.txt. All the graphs show a trend in run time, with an error rate of 0.5 being the highest and 0 and 1 being the lowest. This is most likely due to decoding errors and cache. At 0 there are little errors so decoding does not take a long time due to the fact that most non error code will be cached as soon as they generate. Same with error rate of 1. Most of the Hamming codes have so many errors that they start looking like other valid hamming codes, and thus will become non error cores that are cached. When the error rate is 0.5, half of the codes are valid Hamming codes and half are not, thus the program will need to cache error-free codes, and unvalid Hamming codes with error, thus resulting in roughly two times the amount of time required by error rate 0 and 1, as shown by the graph.

**Entropy Before Encoding:**




This graph, while boring, does show that the entropy of a file before encoding does not change. The difference Entropy values also show that each file has an inherent entropy value different from other files. The complexity of the file determines the starting entropy value.

**Entropy vs Error Rate**




This graph shows the Entropy of a file after it has been encoded and injected with errors. While the files themselves have inherent entropy values, it is very clear that the entropy values rise until 0.5 error rate, then drops at the same rate it rises until it reaches its original entropy. This makes sense because entropy is how uncertain the computer is about what a random byte from the file is. When the file has little errors, the computer is very sure, thus low entropy rate. However, as errors increase, the computer becomes less sure. However, as the error rate approaches 1, almost all the erroneous hamming codes start to turn into valid hamming codes due to the amount of errors, the computer "mistakenly" believes that it knows what a random byte is due to a falsely valid hamming code. Thus the entropy rate goes down again.