Created by Tom Burns and Andrew Dulichan
04/01/2019
CS 214 Systems Programming
Asst2 README

Our Asst2 program was based around a clever implementation of the Huffman Code concept. It deviated a bit from the approach described in the Asst2 project file: we did not use a min-heap and standard tree implementation. Rather, we coded a linked list of binary trees that did the same job. Both of our data structures were of O(n). The insertion was O(n) and sorting was done via the merge sort algorithm which is O(log(n)). This ends in a time of O(n) overall for our total program.

Our binary tree was constructed by repeatedly inserting the node back into the list and then sorting it. The trees were sorted based off the total count value and it did the same job as the min heap and tree approach that the Asst 2 project file stated. Our tree holds both the list of words and their occurrences as trees of those values because the main tree is built at the head of that list. Our program was relatively straightforward and gets the job done well.

For the file testing we did for our test cases, see our testplan.txt file. One thing to note is that we implemented a recursive travel for our test cases which works for directories; see our level0 folder and associated contents for details.