Sam Durst, sd922
Algorithms and Data Structures
27/08/18

Assignment 1 Write Up

The general description of my overall strategy is as follows:

- Reading all the text into an input file
- Stripping all text of punctuation, making the text lower-case, and splitting the words by the default space
- Creating a dictionary with the word as the key, and a count of the word as the value
- If the word had already been seen just add a +1 to the count without re-adding the word
- Adding the value then key, respectively, as a subset to the already created array for the list of words. E.g. [[Count, Word],…]
- Passing the then created array of words and counts to my mergeSort algorithm
- My mergesort was fairly standard, however instead of comparing the values by Array[i], the values were compared at Array[i][0] to check which had a higher count, and if the count was equal, Array[i][1] to arrange alphabetically.
- Print out the first and last 10 values
- Display the time it took to run the program in seconds

The data structures I chose were a multidimensional array and a dictionary. The dictionary is used first to group together the text with its count. The array is then used to hold these values and then be sorted. I chose these data structures because they seemed to be the most efficient options.

The only algorithm I used was a mergesort algorithm. I used this because in the lecture, this algorithm made sense the quickest to me as well as it being very consistent with its worst and best cases. I implemented other functions into my mergesort algorithm so it would alphabetize the text if their count was the same.