

Priyanshu Sethi
12310518
HW - 1
CS 201

Input File	# Of Words	Time Taken
input1.txt	20	0.008381 seconds
input2.txt	9895	5.79685 seconds
input3.txt	19712	11.3049 seconds

The amount of time taken to execute is increasing linearly based on the number of words. It takes a negligible amount of time to execute the program with 20 words but keeps increasing as we increase the number of words. This is correlated to the time taken to perform merge sort which accounts for a significant chunk of this program.

I used an unordered map also known as hashmap for this program since we were required to calculate the frequency and store it alongside the given word before performing further operations on it. Hashmap is the best for such situations since it helps use store key value pairs and access them in the easiest way possible which makes it ideal for this question. Furthermore, I used merge sort to implement the sorting portion of the program since the it has a time complexity of $N \log(N)$ for the worst case scenario. Other aspects of the program take linear time which is added to $N \log(N)$ which would be $N + N + N \log(N)$ in the worst case but they are insignificant for big O analysis for time complexity and hence the over time complexity for this program is $O(N \log N)$