Lexicon Algorithm Report

Algorithm Used: Bubble Sort, Merge Sort

Explanation: Bubble sort is a very basic sorting algorithm to sort a sequence of objects in an ascending order. It also can be used to sort array of strings. In this assignment, an Array List of Strings is used to store all unique words from the input files. Then, bubble sort algorithm is used as a default algorithm to sort the unique words alphabetically. The algorithm examines each element, in our case each word, with its adjacent word. Whenever, a word is not in order, it switches position. The algorithm then repeats switching words until it can run through the entire array list and find no two words that need to be swapped. End of the process, we can get an alphabetically sorted array list.

An alternative sorting algorithm called Merge Sort is also used in this assignment. This algorithm follows divide and merge rule, and uses recursion mechanism. The algorithm first divides the word list in two equal halves and keeps dividing until the smallest portion which can be easily sorted. Then it reverse back by merging two small pieces and sort them in order. End of process the whole list gets sorted in order.

Reason: The main reason behind selecting these two sorting algorithm is to observe their performance to solve a problem as they are different in terms of time complexity and memory usage. Although, in a small dataset, the difference in their performance will not be significant. The Bubble sort algorithm has to go through every elements at least n times to check and swap each element. That means it has n times n. On the other hand, merge sort algorithm, divides the list from the beginning. That means it is always reducing the time complexity by half. That's why it has time complexity of nlogn.

Time Complexity:

Algorithm => => Cases	Bubble Sort	Merge Sort
Best Case	O(n)	O(nlogn)
Average Case	O(n2)	O(nlogn)
Worst Case	O(n2)	O(n)

Execution time:

Test Case	Bubble Sort (Execution time in millisecond)	Merge Sort (Execution time in millisecond)
1	51	46
2	49	45
3	46	44
4	54	49
5	54	51
Average	50.8	47