

Lexicon Algorithm Report

Algorithm Used: Merge Sort

Data Structures Used: Arrays (fixed arrays have $O(1)$ for all operations), ArrayList (only used to insert at last index $O(1)$ and search for existing words $O(n)$)

Explanation: 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: 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 $n \log n$.

Time Complexity:

Cases	Merge Sort
Best Case	$O(n \log n)$
Average Case	$O(n \log n)$
Worst Case	$O(n)$

Techniques Used: As the program designed and written in order to execute it in an efficient time, there are couple of tricks and techniques used in the code.

1. Words from the text file is read, processed and sorted at the very beginning of the program. Because we want to do rest of our operations with clean and ordered words. From that, a list of unique words and their frequency is counted and stored in variables. These processes require a lot of calculations, but the code made sure these processes require minimal time and resources
2. All variables are initialized only once to make sure there is no extra variable. Same variable are used over and over again to make sure there is no extra usage of memory. Less memory usage accelerates the program to execute faster.
3. Fixed length arrays are used mostly in the program as it is the fastest and most simple way to store sequential data. Only one array list is used to find out the unique words as it required less calculations and checking.
4. Unique words having the same length are stored as potential neighbors because we don't want to iterate through the whole word list to find out neighbors for each word. Potential neighbors are stored in an array as comma separated strings whereas array index indicates the length of the word. Later, this string of same length words are used by splitting them with comma and matching them character by character. If only one difference in the characters they are saved as neighbors.