PA2

The data structures being used in this program are fairly simple, two 2-D Character arrays and one node type structure. Each array would hold the list of data words and dictionary words and the node structure would hold the dictionary word with its respective count and superwords.

The complexity comes from the sorting and searching algorithms. The time complexity of my algorithm is O(m^2logm) + O(k^2logk) worst case since I use binary search and insertion sort to enter each word into the 2D arrays for both data and dictionary files and the worst case would occur when each word must be inputted at the beginning of each 2D array. However, the average case would be O(mlogm) + O(klogk). Those are the two most intensive processes in my program. Then I would run another binary search to see how many matches and superwords there were for each given dictionary word. This would all populate a result node which I would later output to a file. The space complexity of my program is a lot less which is at O(n * k) since I only create one data structure of nodes to hold the results for each dictionary word and create two arrays for all the data.

A challenge I encountered during this project was creating an efficient data structure to process all the words in each file. The only potential faster way I could see my program running is by using merge sort instead of insertion sort to process each file and using an array of nodes instead of just character strings.