CPSC 319 Assignment 2

1. Describe the sorting method(s) used in your program – i.e., sorting the array storing all words from the input file (LIST A, step #2) and sorting the letters of two words from LIST A (step #2) to determine whether they are anagrams. Justify your selection of algorithm(s).

The sorting method I used to sort the array storing all the words from the input file is quicksort with the pivot selected from a random number generator. I used quicksort because it has a O(nlogn) and when compared with merge sort requires less memory to run as well as having a lower constant (from lectures). I used a random number generator to select my pivot because that guards from getting an O(n^2) running time from already sorted or reserve sorted arrays ( or almost sorted arrays) it has a high chance of not selecting pivot points that doesn’t divide the data evenly roughly.

The sorting method I used to sort the letters of the words to determine if they are anagrams is an insertion sort method. I choose this because it is a simple sort method O(n^2) as word length will usually be less than the amount of word inputted. It is easier to code and more efficient than bubble sort a similar O(n^2) method and has around the same constant as selection sort.

2. Let N be the number of words in the input word list and L be the maximum length of any word. Give an estimate of the big-O running time of your program. Justify your answer.

To estimate the O time, I calculated the O for each of the steps of my program. For reading the input we have a 4n + 3 because of the while loop which calls four steps (two comparisons, read line and add word to list) this is O(n). For the sorting method for anagram which is called once we have O(L^2) from the known properties of insertion sort. For the quicksort algorithm we have O(nlogn) from the known properties of the quicksort. Then, when I create the linked list from the sorted word list we have O(n^2) as it is a nested loop configuration where the outer depends on n and the inner depends on the outer variable. Therefore, the O(n,L) = O(n^2) + O(L^2) as these are the terms that will asymptotically dominant the expression.

3. What is the big-O running time when the input word list contains only two words?

When our list contains only two words everything with regards to n becomes constant and the only thing that O depends on is L the word length. From the expression we derived O(n,L) = O(n^2) + O(L^2) we see that we have O(L^2) as the O(n^2) becomes constant.