

# CPSC 319

Assignment 2 - Answers

Ethan Reed - 30123128

**Question 1: Worst-case complexity of isAnagram method**

The isAnagram method first checks to see if the input strings are the same length, converts the input strings into char arrays, sorts them using an insertion sort method called charSort, and then iterates through the two arrays to see if they are equal. The time complexity of the toCharArray method used to convert the string is always  $O(k)$ , where  $k$  is the number of characters in the string. The worst-case complexity of an insertion sort is  $O(k^2)$ . The worst-case classification of the final comparison loop exists when the sorted char arrays are equal, in which case the loop must iterate over all  $k$  elements in the arrays. Even in the worst case, the time complexity of the comparison loop has only an  $O(k)$  classification. Because all smaller terms are consumed by the big-O notation, only the  $O(k^2)$  classification of the call to charSort remains, and the overall worst-case classification of the isAnagram method is  $O(k^2)$ .

**Question 2: Worst-case complexity of program**

The worst-case big-O classification of the program exists when the strings in the input file are all anagrams of each other. Calling the sort method will call the linkedSort method once, which uses a modified insertions sort algorithm to sort the strings into alphabetical order. The linkedSort algorithm calls the insert method every time the outer loop runs, and the insert member has a  $O(N)$  classification,  $N$  being the number of words in the linked list. The inner loop of the insertion sort algorithm also has an  $O(N)$  classification. Because coefficients are absorbed by big-O notation, the time complexity of the inner loop is just  $O(n)$ . The outer loop of the algorithm runs  $N$  times, once for every word, so because of the product rule, the overall classification of linkedSort method is  $O(N^2)$ .