Name: Paul Murariu

Teacher: Professor Jinhua Guo

Course: CIS 350 – Section 102

Date: 5/29/2022

Description of the problem in own words: This problem tasks us with creating a spell checker that consists of three command line arguments which are specifically the number of words in the dictionary, a dictionary file name, and a text file name. The program that is created will produce a hash table, then read off the dictionary from the specific file, then insert the words into the created has table, and finally report the statistics of the collisions. Once the program has read the dictionary, the created spell checker will read a list of words from a created text file. Each specific word will be looked up in the dictionary and if the results end up being incorrect, the output will end up reverting to a standard output combined with a list of suggested corrections. Finally, you also need to find the time analysis worst case O() of each function with explanation, including the main function, and the average case O() of the contains, insert, remove, and findPos functions as well.

**Time Analysis Cases For Member Functions With Explaation**

**QuadraticProbing.h Functions**

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

Average case complexity is: O(1)

The average case complexity is O(1) because the function has only one statement that is running.

Text

Description automatically generated

Worst case complexity is: O(n)

Text

Description automatically generatedThe worst case time complexity is O(n) because the function has a for loop, which runs for n times.

Worst case complexity is: O(1)

The worst case time complexity is O(1) because in the function, insertion in HashMap takes O(1) time.

Average case complexity is: O(1)

The average time complexity is O(1) because the length of the function is constant, and so as a result, inserting is done in constant time.

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because in the function, the deletion in a HashMap of a specific key will always be a O(1) statement.

Average case complexity is: O(1)

Text

Description automatically generatedThe average case complexity is O(1) because the function is only returning one statement.

Worst case complexity is: O(n)

Text

Description automatically generatedThe worst case time complexity is O(n) because the function has a for loop, which runs for n times.

Worst case complexity is: O(1)

The worst case complexity is O(1) because the function has only one statement that is running.

Text

Description automatically generatedWorst case complexity is: O(n)

Text

Description automatically generatedThe worst case time complexity is O(n) because the function has a for loop, which runs for n times.

Worst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

Text

Description automatically generatedWorst case complexity is: O(1)

Text

Description automatically generatedThe worst case time complexity is O(1) because the function has only one statement that is running.

Worst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only statement that is running.

Text

Description automatically generatedWorst case complexity is: O(n)

The worst case time complexity is O(n) because the function has a while loop running at O(n).

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because in the function, searching for a key in HashMap takes O(1).

Average case complexity is: O(n)

The average time complexity is O(n) because the function has a while loop, which runs for n times.

Text

Description automatically generated

Worst case complexity is: O(n)

The worst case time complexity is O(n) because the functions has a for loop that runs n times.

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

**QuadraticProbing.cpp Functions**

A screenshot of a computer

Description automatically generated with medium confidenceWorst case complexity is: O(n)

The worst case time complexity is O(n) because the function has a for loop, which runs for n times.

A screenshot of a computer screen

Description automatically generated with medium confidenceWorst case complexity is: O(n)

The worst case time complexity is O(n) because the function has a for loop, which runs for n times.

Text

Description automatically generatedWorst case complexity is: O(n)

The worst case time complexity is O(n) because the function has a for loop, which runs for n times.

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running.

**SpellChecker.cpp Functions**

Text

Description automatically generatedWorst case complexity is: O(n^2)

The worst case time complexity is O(n^2) because the function has a nested for loop which runs for n^2 times.

Text

Description automatically generated

Worst case complexity is: O(n^2)

The worst case time complexity is O(n^2) because in the function, inside the while loop there is nested for loop for doing operations.

Text

Description automatically generatedText

Description automatically generatedWorst case complexity is: O(n^2)

The worst case time complexity is O(n^2) because the function has a nested for loop which runs for n^2 times.

Text

Description automatically generatedWorst case complexity is: O(n)

The worst case time complexity is O(n) because the function has a while loop, which runs for n times.

Text

Description automatically generated

Worst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only one statement that is running. The function is simply just outputting.

**TestQuadraticProbing.cpp Function**

Text

Description automatically generatedWorst case complexity is: O(1)

The worst case time complexity is O(1) because the function has only statement that is running.