

CS260 - Data Structure

Programming Assignment 3: Spell Checker

Assignment Report

Fitsum Alebachew fa496

Question 1 : 8 points

Data Structure: Answer the question below regarding the data structure you used in your implementation:

- (a) [2 points] **Overview of Hash Table:** Briefly explain the hash table you used to store the dictionary on memory. What was the advantage of using an open hash table over a closed hash table?

Solution: It is an open hash table that uses linked lists in each hash value address. It is better than a closed hash table in its ability to store more than one element with the same hash value.

- (b) [2 points] **Hash function:** Given a single word as input, how did you define your hash function? Briefly explain its behavior.

Solution: The hash function uses the char values of the characters and adds them up by multiplying by 101 before adding consecutive characters.

- (c) [4 points] **Statistics about your Hash Table:** How large of a hash table did you use? How many times did collisions happen when you were loading the dictionary initially? What was the length of max/min/average linked list in your Open Hash Table? (refer to the last slides of lecture 08_hashTables.pdf for an example statistical analysis of a hash table)

Solution: I used the number of words in the dictionary as the length of the hash table. The expected average number of elements in a row is 1.
Min: 0, Max: 2

Question 2 : 8 points

Searching: Answer the question below regarding how you searched for words in the hash table:

- (a) [4 points] For the words that you couldn't find in the dictionary, how did you check for alternative spellings for the three types of common errors that you were asked to check for?

Solution: I checked for each specified case separately by adding/deleting/shifting letters in the input file words and seeing if they are members of the hash table.

- (b) [2 points] In terms of the size of the input text defined as n words and dictionary size defined as m words, what was the **time complexity** of spell checking for the entire input text **to identify mistyped words**?

Solution: This is n times the time complexity of searching for one word which is $O(N/S) = O(1)$. This comes to be $O(n)$.

- (c) [2 points] With n and m defined same as above, what was the time complexity of spell checking **for a single word to suggest alternative spellings**?

Solution: This checks multiple cases for each word. But since it is multiplied by a constant, the time complexity is still $O(1)$.

Question 3 : 4 points

Overall:

- (a) [2 points] Do you think your program could have been improved? If so, how? What would improve: runtime, memory usage, both? Considering the statistics you calculated on the first problem, what could be a good balance point for the size of your hash table for an optimal memory usage and runtime?

Solution: The size of the hash table gives it a fast running time, since collisions will be minimized. However, this increases the space complexity so a better ratio between the length of the table and the number of words being inserted will be better. The ratio might depend on our need for better space vs time complexity.

- (b) [2 points] If you were to implement a real life spell checker, how would you do it? That is to say, in practical use of a spell checker, what could you do to offer a more efficient program to the end user?

Solution: More cases should be checked like missing characters in the middle of the word, or added characters in the middle of the word.