Jack McVeigh

CS 260

22 October 2021

Programming Assignment 2: Spell Check Report

**Introduction**

Spell check is a well-known feature in all forms of textual user input, whether it be in texting, emailing, document editing, or programming via an IDE, spell check provides a helpful oversight when inputting thoughts into a keyboard. Spell check can be implemented in many ways via a hash table, and depending on the algorithms used in the process, spell checkers can provide near perfect recommendations to a user. The chosen implementation for a spell check program can be observed in the below flow chat, Figure 1.

Diagram

Description automatically generated

Figure . Spell Check program execution.

**Loading the Hash Table**

The hash table is loaded from a file by reading the file line by line, hashing each word and pushing the word to the correct linked list based on the hash value. Figure 2 contains the function that runs this process.

Text

Description automatically generated

Figure . Loading an open hash table via a file.

The size of the hash table is dynamically determined via the process in Figure 3. The hash table’s size grows logarithmically to provide higher suggestion hit rates as the size of the input dictionary grows.

Text

Description automatically generated

Figure . Determine the size of the hash table.

Figure 4 contains the hashing algorithm selected for this implementation.

Text

Description automatically generated

Figure . The hashing function. String score is the sum of each of a word’s character's ascii value divided by its length.

**Determine if Input is in the Hash Table**

Searching the hash table for the given word is as easy as just hashing the word and searching the corresponding linked list for the word. If the word is in the linked list, the word is a correctly, typed word. If not, the word is mistyped.

Text

Description automatically generated

Figure . Searching the hash table for a given word.

**Determine Suggestions for Misspelled Word**

Finding suggestions for the user is done by taking the hash value of the input word and manipulating the corresponding linked list’s words to match the input word. If a word in the dictionary can be manipulated by the means of swapping letters, removing letters, inserting extra letters, or replacing existing letters, then the candidate word can be provided to the user as a suggestion. Figure 6 shows this process.

Text

Description automatically generated

Figure . Manipulate word in corresponding linked list to find suggestions.

After the suggested words are determined and if any are found, they are presented to the user. Figure 7 shows the output of the whole process.

Text

Description automatically generated with medium confidence

Figure . The output of the process of finding suggestions.

**Ask User to Write Mistyped word Back to the Hash Table**

Before continuing the program to prompt the user for another word, first the program asks the user if they would like to include their input word in the dictionary.

Text

Description automatically generated

Figure . Writing input word back to the hash table.

If the user opts to write the word back to the hash table, subsequent runs will be able to check against this newly added word.

**Conclusion**

The running time of the spell check algorithm implemented in this project for correctly spelled words is 0(n), where n is the length of the corresponding linked list for the hashed input word. The running time of the algorithm for mistyped words is O(n^2) as many more operations must be done to determine the suggestion words. To better implement this algorithm and to obtain better results with the same time complexity, a better scoring system for suggestion words must be developed. This implementation lacks that capability due to time constraints.