Thomas Jorgensen

Programming Assignment 5

Spellchecker with Binary Search Tree

November 8, 2015

Spellchecking with Binary Search Tree

Spellchecking in programming can be done in multiple ways. You can do it recursively, or use an array, or linked lists. Another way of doing it is by creating a binary search tree. Binary search trees are objects that allow for you to find an item quickly because you are searching both higher and lower at the same time. This allows for objects to be created and the object oriented principles able to be used.

This program entails the use of reading a file and a parser. It also creates a stream of information to be used. In the binary search tree it uses the current that is created by the nodes to search via the first letter in the alphabet. The dictionary is created based on the first letter of the alphabet and put into an object of binary search tree. This dictionary is then used by another write parser method. In this method the words are searched by the document and determined whether a word is correct or not.

The point of the program was figure out how many comparisons are used on words that are found and words not found to see if there is a correlation between how many words it has to go through or if it has to go through the entire list. The results of this program show that there are 54648 misspelled words, and 937492 correct words spelled. The average number of comparisons for correct words is around 10.40 comparisons per word and the average number of comparisons for incorrect words is around 16.35 comparisons per word. This shows that the average number of comparisons for correct words and incorrect words is a ratio of 1 to about 1.5. This indicates that there is not a huge difference between the misspelled words and correct words.

Doing this program a few different ways now shows that a comparison can be determined on the best method, and a reason as to why. The first method used recursive binary search. This method was slow and was linear. The next method was my Linked List method. This method was not super slow but the average number of comparisons was in the thousands not the tens. The last method was a binary search tree. This method was not slow at all and was accomplished in a matter of seconds not minutes or hours. This method was fast due to the fact that not a lot of comparisons was made and so because the computer had less work to do it could accomplish it faster.

Output:

```
run:
Number of misspelled words: 54648
Number of correct spelled words: 937492
Number of misspelled words comparisons: 568211
Number of correct spelled words comparisons: 15325631
The average number of comparisons for words found: 16.347479231822778
The average number of comparisons for words not found: 10.397654077001903
BUILD SUCCESSFUL (total time: 21 seconds)
```