Quinten Bailey
Programming Assignment 5
Spellchecking with Binary Search Trees
11/8/15 11:59 PM

Spellchecking With Binary Search Trees

Quinten Bailey, Computer Science (Class of 2017)
Research Advisor: Dr. Suranga Hettiarachchi

Spell checkers are present in nearly every word processor available today. It is a tool that many people have come to rely heavily upon. This program is similar to Program 2 and 4, but instead of searching recursively or with Linked Lists, This program uses Binary Search Trees (BSTs). An array of BinarySearchTree objects is created and the randomized dictionary is loaded into this array one word at a time based on the first letter of the word. The ASCII value of the first character is used to determine which list to place the word into. The file being spell checked is then fed into the array one word at a time, again using the ASCII value of the first character of each work to search in the appropriate BST. The program needs to search less than Program 2 because all of the "a's" are in one list, the "b's" in another, and so on. It takes less time than Program 4 because its structure allows for quicker navigation. Unlike a LinkedList, the BST will have to examine far less elements to find a match. This means that the average number of comparisons required to either find a match or determine the element is not in the list is far less than either of the previous data structures we have used. In addition, the time complexity is better using BSTs opposed to LinkedLists.

```
/*
Number of words found: 937492
Number of words not found: 54648
-------------------------------------
Number of comparisons for words found: 14388139
Number of comparisons for words not found: 568211
-------------------------------------
Average number of comparisons for words that were found: 15.0
Average number of comparisons for words that were not found: 10.0
*/
```