Samuel Bohannon

Assignment Five

Spellchecker with Binary Search Trees

Due on November 8th 2015

Description of the problem: In this assignment, another version of the spellchecker will be implemented. A random dictionary text file is given and is to be read into an array of Binary Search Trees organized using the first letter of each word. Then, a book file (oliver.txt) is to be read, and each word pulled one at a time. These words are to be searched for in the dictionary array. If a word is not found, it is potentially misspelled and counted, if it is found then the word is correct and is also counted. The averages of words found, words not found, and average number of comparisons for words found and words not found should be output in the final program.

Algorithm and Design: First an array of Binary Search Trees is constructed using the most recent lab files. The dictionary file is read and the data is inserted into a tree in the array depending on the first letter of the word. 'a' = 0, 'b' = 1 and so on. The book file is then read one word at a time, and each word is sent into the array of trees as an attempt to find it using the search method written into the Binary Search Tree class. If the word is found, add one to the Words Found counter and add the value of the comparison counter to the counter corresponding to words found, if the word is not found, add one to the Words Not Found counter and add the value of the comparison counter as before. The counters are then output as well as their averages.

Observations and results: This program ran much faster than Assignment 4, and a bit faster than Assignment 2. Assignment 4 took about 45 seconds to fun, Assignment 2 took about 14 seconds, and Assignment 5 took about 8 seconds. This is due to the fact that the number of comparisons for failed searches in this case is actually *lower* than the number of comparisons for a successful search. For Assignment 2, there were around 16 comparisons for the successful search and 17 for a failed one. With this assignment there were again around 16 comparisons for a successful search, but only around 10 for a failed one.

Output:

Number of words found: 937492

Number of words not found: 54648

Average number of comparisons for words found: 16.35

Average number of comparisons for words not found: 10.40