Farhad Amiri
Programming Assignment 5
Spellchecking with My Binary Search Trees
Due Date: 7/21/2015

The main goal of spell checker program is to analyze documents and find words that might be misspelled. The program works by comparing each word in the document to a large dictionary of words. If the word is not found in the dictionary, it is flagged as potentially incorrect. It is required to count and print out; the number of words found, number of words not found, number of comparisons for words found, number of comparisons for words not found , average number of comparisons for the words found, and average comparisons for words not found.

Additionally, the program should read dictionary text file into an array of 26 Binary Search Trees, one for each letter of the alphabet. Consequently, the first BST would contain only those words starting with the letter 'a', while the last would contain only those words starting with letter 'z'. Then, when the program reads in the book, it will examine the first character of each word, and search the word in one of the BST. If it is not found, then output the word. This word is either mis-spelled, or not in the dictionary. The words in the dictionary and the book should be run through the String Parser.

In this technique, there are 5 methods (Reading dictionary method, Search word method, Read Oliver method, Main method, result method) which are responsible for separate individual tasks. For instance; the program will read and store the dictionary text file into an array of Binary Search Trees by using read file dictionary method. Then the Boolean linear search method will search for the words and analyze with the Oliver text file method. The main method will be implemented, which will call all methods.

In conclusion, we will see the program result by the result method. The Program result shows us complete information about the text file which helps us to find errors in text file and how the Binary Search Trees compared words. In addition, the program counts the number of incorrectly spelled words; total number of recursive steps for all incorrectly spelled words, and output these counters along with the average number of recursive steps per incorrectly spelled words.

Finally, the results of three assignments are different because we used different algorithm structures to store or search. For example, in the program assignment two the number of average comparison is almost close to assignment five but assignment four is the largest. Because in the assignment two, we used binary search algorithm, and we stored our files in an array. In assignment five, we stored our file into a binary search tree array, but in assignment four, we stored into mylinkedList and used linear search. It takes more time to do the process and find the result. As the results show, the best case for these three assignments is the assignment five because it took a little amount of time to find the result.