Hannah Bloyd

Programming Assignment 5

Binary Search Tree Spell Checker

April 11, 2017

This program was designed to test how well BinarySearchTrees work compared to LinkedLists. In the last lab, we created a dictionary and then tested the dictionary with text file Oliver, using LinkedLists. This program is doing the same thing; the only difference is using BST instead. Using the time each program took and the average computations for finding and not finding words, we can compare with method is better.

Using BinarySearchTrees to create a dictionary is significantly more time efficient. Searching the dictionary by means of BinarySearchTrees took 9 seconds to finish the Oliver text, while the LinkedList dictionary took 11 minutes and 4 seconds. Both programs had 912,807 words found, but the BST method had an average of 16 computations to find the word as opposed to 3,550 computations for the LinkedList method. Similarly, both programs had 64,853 words not found. The BST method used an average of 11 computations and the LinkedList method used an average of 7,437 computations. Because the number of average computations is so much higher for LinkedLists, the time goes up drastically as well. BST is faster because it is sorted, unlike LinkedLists. When a BST is loaded, lower values go to the left and higher values go to the right. This means that the time complexity for searching a BST is O(log N) in the worst case. LinkedLists are not sorted in any way, so the worst case time complexity is O(N).

Output:

run:

words found 912807

words not found 64853

Comps Found 14926560

CompsNotFound 742781

Average comps for words found 16.0

Average comps for words not found 11.0

BUILD SUCCESSFUL (total time: 9 seconds)