Jacob Jones

Assignment5

Dictionary Comparisons with Binary Search Tree

11/1/2015

The goal of this program is to compare a large text file to a given dictionary. For each word, we must count how many comparisons are required to find the word in our organized dictionary. The major goal of this assignment was to do the same project as assignment two and four, but with binary search trees instead of a binary search and linked lists. To accomplish this we created a binary search tree for each letter of the alphabet. The dictionary was fed in one word at a time, and the word was then assigned to its correct tree. We then proceed to compare the large text file (oliver.txt) to our linked list dictionary. To do this we would pull in the next word in the text file, and figure out the first letter of the word. With the letter, we would compare the word to the correct binary search tree. The word would make its way down the tree using its value. At the end of the program the counters are displayed showing number of words in and not in the dictionary along with total and average comparisons for each.

Binary Search Trees are much more effective in terms of total searches. Unlike linked lists, binary search trees are organized by value into separate trees, making the end values can be reached much easier. Because the tree divides up into many branches, there are less total steps to reach any of the possible values. This led to drastically decreased comparisons and much shorter runtime than linked list. Linked list must iterate through each value of its list to discover that what it is searching for does not exist. Binary search tree on the other hand, only needs to continue down the path in which a value might exist. This allows it to not compare to every value in the tree and save time. The average number of comparisons also seems to be backwards in binary search tree. To find a word actually took longer than realizing a word did not exist in the list.

Output:

```
run:
Words found/correct words: 940320
Words Not Found/incorrect: 59221
total comparison: 15295584
word incorrect comparison: 568198
avg comparisons for words found: 16.0
avg comparisons for words not found: 9.0
BUILD SUCCESSFUL (total time: 4 seconds)
```