Kendra Driver and Emma Romig
Program 5
Assignment Five: Comparisons
July 21, 2015

Programing Assignment 5

Kendra Driver and Emma Romig, SU15-SE-CSCI-C202-3183, Java

Assignment Five required the programmers to implement a second version of the existing spell checker. The main goal of this assignment was to create several spell checking methods with different program styles and compare the results of each. Program Two, Program Four, and Program Five was required for this project. For all the programs Oliver Text and Dictionary Text was used. Each program also contained a counter. This counter would keep track of number of words found, number of words not found, number of comparisons for words found, number of comparisons for words not found, and average of number of comparison for the words found and average comparisons for words not found. Once all of the counters were created and supported the programmers were able to make a decision on how the different programs differ and which seems quicker and most efficient. The results were interesting. Program Two ran well. It did the job and it worked quickly. Program Two was a good running assignment; results were notable. The amount the timer output was 12.7 seconds. Program Four ran the least impressive of all. Program Four worked the way it was coded and had a time of 70.9 seconds. Nothing was wrong with the run time, however compared to the other two programs it preformed the slowest. Program Five ran the most efficient out of the three programs tested. Five had a run time of 9 seconds. It was least expensive of all. Comparing the programs was interesting. The time taken to solve the same problem using Linked Lists was much longer than using Binary Search Tree method. The reason why Binary Search Tree's are more time efficient is because the search space is being cut in half every time a search happens. For Linked Lists the search has to go through the entire data set. For large data sets, searching one with a Linked List, will take a lot of time. However with Binary Search Tree's the data set is cut in half. After each cut the list gets smaller and finding the data becomes quicker. Overall, we discussed that Program Four was the slowest, Program Two was noteworthy, but Program Five is the most efficient program tested. Binary Search Tree's are quicker than Linked List searches.

Comparisons and Times

Program Two:

```
run:
~~Recursive Binary Search~~
Found: 895738
Not found: 96402
compCountF = 29935066844211
compCountNF = 66970128
Average found recursive: 33419445
Average not found recursive: 694

Time elapsed: = 12.651333
BUILD SUCCESSFUL (total time: 12 seconds)
```

Program Four:

```
run:
~~MyLinkedList Search~~
Found = 939673.0
Not Found = 52467.0
compCountF = 3289720729
compCountNF = 388329531
Total = 992140.0
Average comparisons for words found = 3500.920776695723
Average comparisons for words not found = 7401.405283320944

Time elapsed: = 70.791768
BUILD SUCCESSFUL (total time: 1 minute 11 seconds)
```

Program Five:

```
run:
~~Binary Search Tree~~
Found = 939673.0
Not Found = 52467.0
compCountF = 6103795026
compCountNF = 369482578
Total = 992140.0
Average comparisons for words found = 6495.658623797853
Average comparisons for words not found = 7042.189909848095

Time elapsed: = 9.413507
BUILD SUCCESSFUL (total time: 9 seconds)
```