Ryan Madden

Assignment Five

C-202

Spell Checker Binary Search Tree

Programing assignment five required a reimplementation of Assignment Four's linked list spell checker. Aside from a few differences in the methods of our binary search tree and the my linked list classes the spell checker was implemented identically

An array of binary search trees was created with 26 indices to hold words read in from a dictionary file, with each starting letter given its own list. Index 0 contained "a" words while index 25 held words starting with the letter z. Using ASCII values each word read in was appended to the appropriate linked list. Once the dictionary was created, a text file, the novel Oliver was read in line by line. Each line was trimmed of white space, its characters were converted to lower case and using regular expressions, all undesirable character values were replaced with white space. The string was then split by its white spaces and each element of the newly created string array compared to the dictionary file, using search method created in binary search tree class, element by element. For all elements, the path method was executed to create an array from the tree root to word in question. If the word was in the dictionary this returned arrays length will show us the number of comparisons made to find it. To find comparisons not found in dictionary a new path method was created called path2. The original requirements of the path method needed to return an empty array if an element was not found in our data structure. Path2 does not clear the returned array because its size shows us the comparisons made in our binary search tree if the word was not present in out dictionary file.

Results: (Original) Linked List

Correctly Spelled	940250
Misspelled	059291
Avg. Comparison of Found	3558.37
Avg. Comparison of Not Found	7903.53
Time converted from nanoseconds	54.199354709

Results: New Binary Search Tree

Correctly Spelled	940258
Misspelled	059283
Avg. Comparison of Found	16.34
Avg. Comparison of Not Found	9.66
Time converted from nanoseconds	2.086132126

The tables above show correctly spelled words (words found), misspelled words (words not found) and the average comparisons of each. Average comparisons show the number of checks each category completed in attempt to locate a given word in the dictionary array. I cannot account for the difference, in the eight additional words found to be correctly spelled between the binary search and linked list. The string parser was not changed between testing.

The significant difference between the data is average of comparisons and execution time. The average comparison numbers show the work that was done to determine if a word was correctly spelled for each data structure. The extreme difference between the two data structures shows the great efficiency of a binary search tree for this task and highlights the weakness of large data traversal with linked lists. The execution times of each program reflect the same data and show the real word differences between O(long(n)) and O(n).