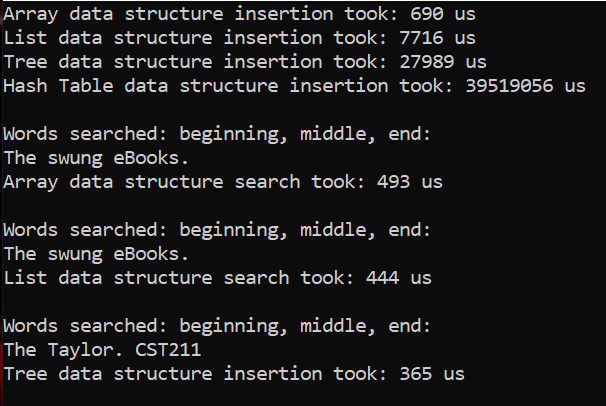
Timed Insertion/Search for All Data Structures



As far as my results go, the insertions look about correct in that an array data structure should be very fast in insertion vs the other data structures Big-O notation. However, I found it a bit odd in that my Hash-table was extremely slow. I’m not sure if it how I implemented my hash-table or something else. This looks like an outlier and makes this data subset not seem too true. So, I am skeptical on the hash table time but other than that, it looks about right. The search times seem to be a bit weird as they are constantly all getting about the same amount of time to search for each element. I would think that an array is significantly faster than a linked list in that it doesn’t have to traverse through all the elements to get to the element we are searching for, and the same can be said for the tree data structure. The hash table does seem to be about right since the hash table takes the characteristics of the array and utilizes it. Its still too fast and I think there is something wrong.

I think instead of using our data structures created in class, we should have used the ones that were created via the STL library (asides from the tree data structure since it doesn’t exist). I would also like some more concrete ways in the lab to test the search and insertions of our data structures, since the ways that I’m doing it seem wrong. I would have liked to get more concrete evidence that proves the Big-O ranked data structures are in fact true when comparing data structures.