# RUN TIME PERFORMANCE ANALYSIS

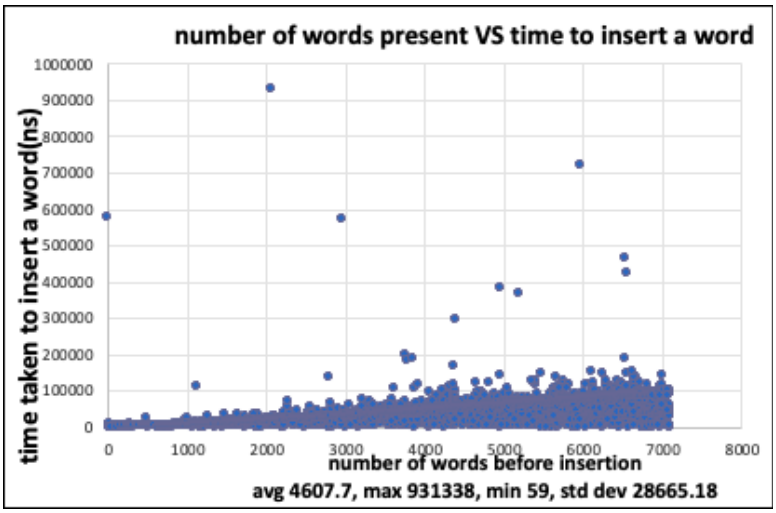
The following are some findings from the runtime performance analysis conducted on all three set implementations.

- Time taken for Adding and Searching on “linked list implementation of set” increased as the number of words present increased. This directly correlates to the insertion runtime of this algorithm which is O(n). On average this implementation took 46068 ns to add a word to the set, and about 17704 ns to search a word from the second file. The theoretical runtime for search is O(n) as well.

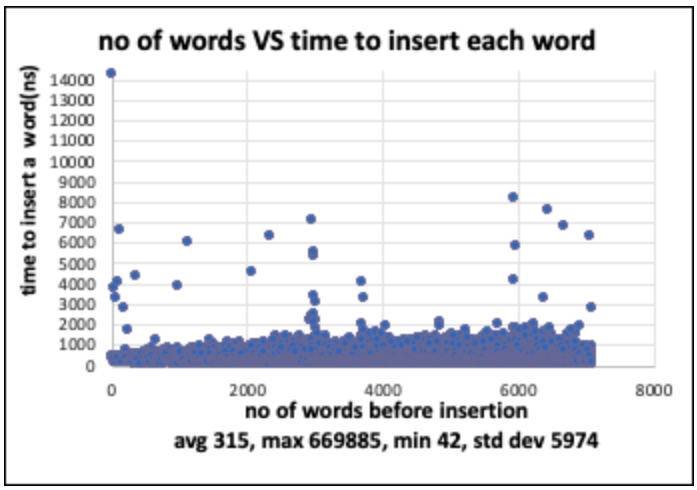
- The binary search tree implementation took only 316 ns on average to add a word, and about 727 ns to search for a word. The theoretical runtime for insertion and search is O(log n) on average and O(n) worst(when the tree becomes a list), n = no of nodes.

- The hash table implementation was the fastest taking only about 182 ns on average to add a word, and about 332 ns to search for a word. The fast runtime stems from the fact that hashing uses a concept where elements are mapped by an index, therefore every time an add or a search is performed it happens in constant time on average. Insertion - O(1), search - O(1).

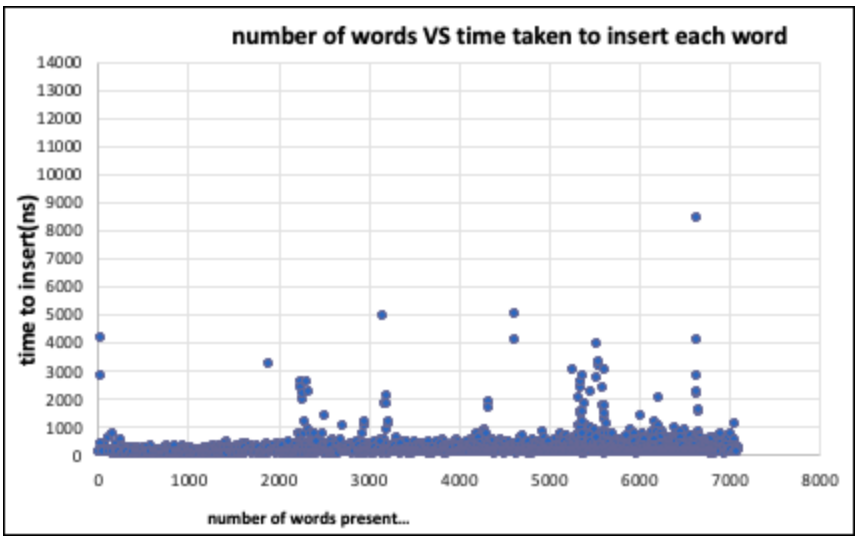
- Search(Linked List set) average- 17704 ns max time 139507 ns, min 177 ns, std dev - 11468.75
- Search(Binary Search Tree set) average 727 ns, max time 131969 ns, min 155 ns, std dev - 5085.5
- Search(Hash Table set) max time -average 332 ns, max time 17954 ns , min 135 ns, std dev - 697.08




Linked List Set




Binary Search Tree Set




Hash Table Set