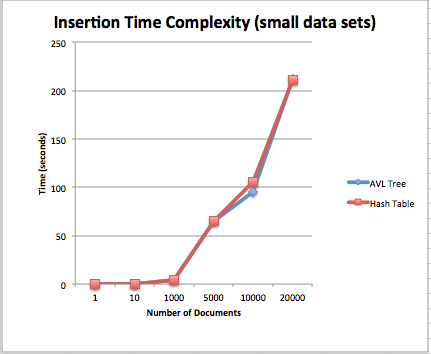
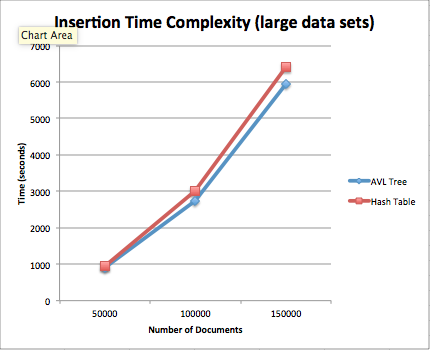
Charlie Barnes, Zach Bubrosky, Wesley Smith

CSE 2341

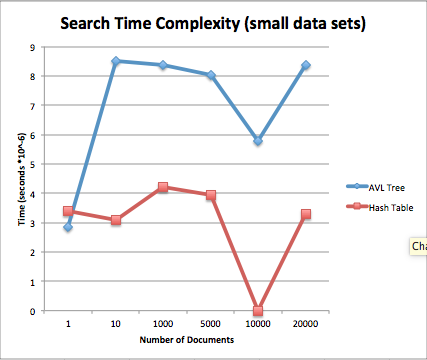
Final Project Paper

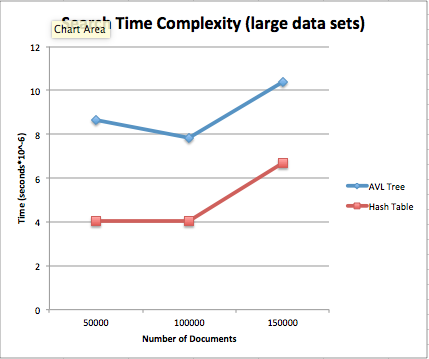
The Hash Table and AVL tree both have upsides and downsides. In small sets of data (less than 10,000 documents), both have relatively the same parse time. However, in larger sets of data (greater than 10,000 documents) AVL parse time was more efficient. In general, AVL parse time was 5-10% faster than Hash Table parse time. Factors behind this could involve that the Hash Table handles collisions by separate chaining using an AVL tree. This would indicate that the additional time used by the Hash Table was because of hashing the word to get its index value and creating a new hash node to be inputted into the hash table’s index’s AVL tree. Additionally, Hash Table’s insertion time complexity is O(n) in worst case and AVL tree’s insertion time complexity is O(logn), thus because the Hash Table is chained with AVL trees so the complexity with collision is O(nlogn).





For search time, the Hash Table proved to be significantly faster in relativity (times only had fractional differences). In general, Hash Table search time was approximately 200% faster in data sets with sizes up to 100,000 documents. Both search times remained approximately the same throughout this sample size range. However, this rate began to slow at a data size sample of 150,000 documents. Even at this slowing rate, the Hash Table still surpassed AVL search by approximately 21% in extremely frequent word (such as “book”). But, for less frequent ranges margins remained consistent with the data set sizes of 100,000 or less. The reason behind the Hash Table having a more efficient search is because its average time complexity is O(1), while AVL tree’s is O(logn).





In comparing the data structures, the Hash Table’s was more efficient at searching, but this time was negligible when including its insertion time because the difference in search times was only a fraction of a second. The AVL Tree was better for larger data sets in insertion time and relatively the same in smaller data set. In conclusion, we found the AVL tree was the better data structure to use for our search engine.