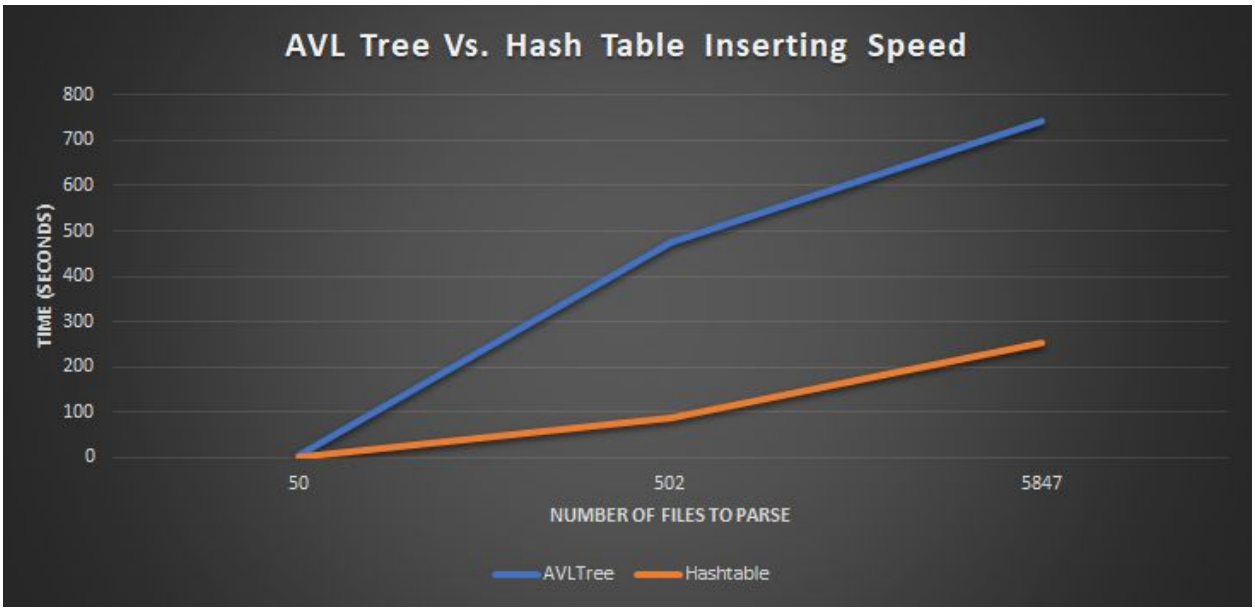
Tim Desmond and Grant Ayers

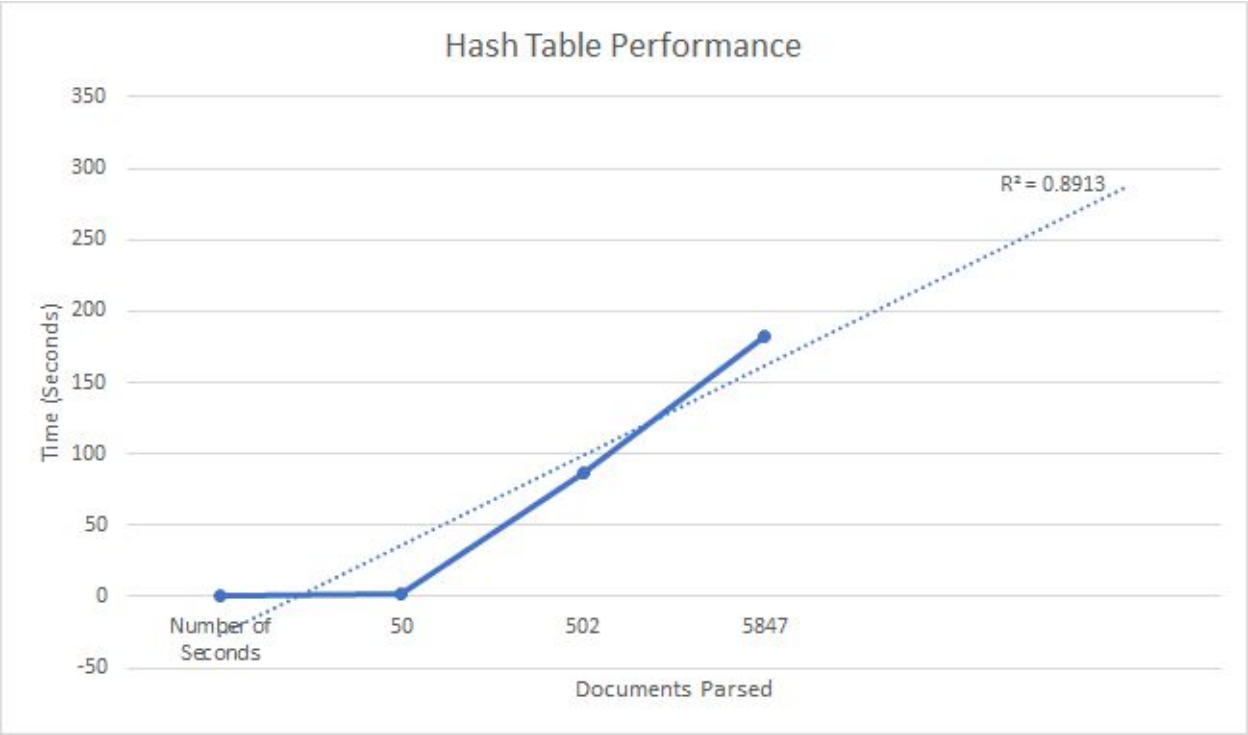
Comparison of our AVLTree and Hash Table

The primary data structures utilized in this project are the AVLTree and the hash table, each with its own benefits and drawbacks. A hash table functions by assigning each value a key which is then hashed. To search for a value in the hash table, the table uses the key to find where the value is stored. AVLTrees are an evolution of an ordered binary tree characterized by not allowing two subtrees to have a height difference of more than one level with rotating functions to rebalance the tree. To compare the two data structures, the time to insert and access will be analyzed.

Our team predicts that the hash table will be faster at inserting and accessing elements due to its ability to run at O(1) rather than the AVLTree's O(LogN) time complexity. We must consider that our implementation of these structures may not be ideal so the time will likely vary from these approximations. To investigate this hypothesis, our team implemented both an AVLTree and hash table in the same program running on the same sets of data. The data structures were tested seven times each on three different data sets to monitor the growth of time taken as the number of documents to parse increases.



(Figure 1)



(Figure 2)

(Figure 3)

The R^2 for the linear line of best fit for the hash table is .89 showing that the trend is strongly in the correct direction. The AVLTree R^2 value of a logarithmic line of best fit is .7793, showing a weaker prediction. As you can see from figure 1, the hash table was faster than the AVLTree at all three tests each time showing that, based on our implementations, the hash

table is across the board faster than the AVLTree. To prove this, we will generate and analyze a two-sided confidence interval with a confidence level of 95% to determine the difference for each of the tests. For the 502 documents set, the mean speed for the AVLTrees lies within 460.81 and 492.12 with 95% confidence which is not even close to the 81.92 to 92.43 second 95% confidence interval for hash tables. This data allows us to reject the null hypothesis that hash tables are not faster than AVLTrees.

Based on our analysis of the data structures, the hash table is faster than the AVLTree. It is worth noting that AVLTrees have other advantages over hash tables such as having sorted data, not needing to perform costly resizing functions, and generally being easier to create and manipulate. AVLTrees are much closer in speed at lower document sizes, but the gap grows wider the larger the document pool. However, when it comes to speed for larger datasets, a hash table is the superior data structure.