The main use for both our AVL Tree class and our hash table is to be able to store terms as they are parsed from the documents into an index. Both data structures have their benefits within the different sections of the program, and for different sizes of data. First the AVL Tree within our program is a templated class capable of storing values in a way that can allow it to be easily searched, due to the way it is structured. This data structure had several key benefits within our program, its ability to be quickly searched and inserted into made it an effective tool for reducing parsing times, by having an AVL tree of our stop words be searched through for matching words in the documents we were parsing. In addition due to the fact that and AVL is already sorted searching for values was very efficient making it a great tool for finding and returning values that may or may not be in the documents. Our AVL tree is most efficient for smaller data sizes often slowing down as it has to deal with larger more cumbersome file sizes. On the other hand our hash table has the advantage that it is able to have elements nearly instantly accessed making it very useful for searches and accessing data. Ultimately AVL Tree was the faster more reliable data type for parsing and storing terms.