

Gisselle Sagastume

CSCI 335

In part of my project, I used an AVL tree and a hash table to create a library book catalog system. The AVL tree is used to maintain data balance and speed of search. It is known that the AVL tree is known to be a kind of binary search tree that maintains its balance, whenever you add or modify part of it will make sure to maintain the balance needed. I made my project so that my AVL tree has nodes that contain a value of the author and genre combined, just as the book title. In my project, the hash table is to be an array, and each spot in the array uses an AVL tree to handle collisions. So that when two or more books hash to the same spot, the AVL tree keeps them balanced and sorted.

I converted each book title into a hash number using my `Hashtable::hash` function. This hash number decides which slot the book goes into in the hash table. Instead of using a linked list, the AVL tree keeps both books in a balanced manner if they hash to the same node. This technique is known as separate chaining, however to maintain speed even when multiple books share a hash node, I used an AVL tree rather than having them in another form. In this project I had to import the file called `books.txt`, search for books by title, view the complete hash table, and exit the application using the simple menu. The lines in the `books.txt` file contain the title, author, and genre of each book, all of which are separated by commas. Because the tree remains balanced, the time complexity for searching or inserting into a single AVL tree is $O(\log n)$. It is interesting to know that AVL will always be $O(\log n)$. If the hash function is functioning properly, the average search time should be $O(1)$ since the hash table distributes the books over several slots. When all the books are added, my application uses $O(n)$ space in total. Through this project, I discovered how to quickly and efficiently store and locate books by combining an AVL tree with a hash table. Collisions are handled more quickly when an AVL tree is used rather than a linked list. This project made it more clear for me to understand how actual systems use a variety of data structures to effectively handle high volumes of data.