Mason Bell
Meriah MacKillip
Jake Werner

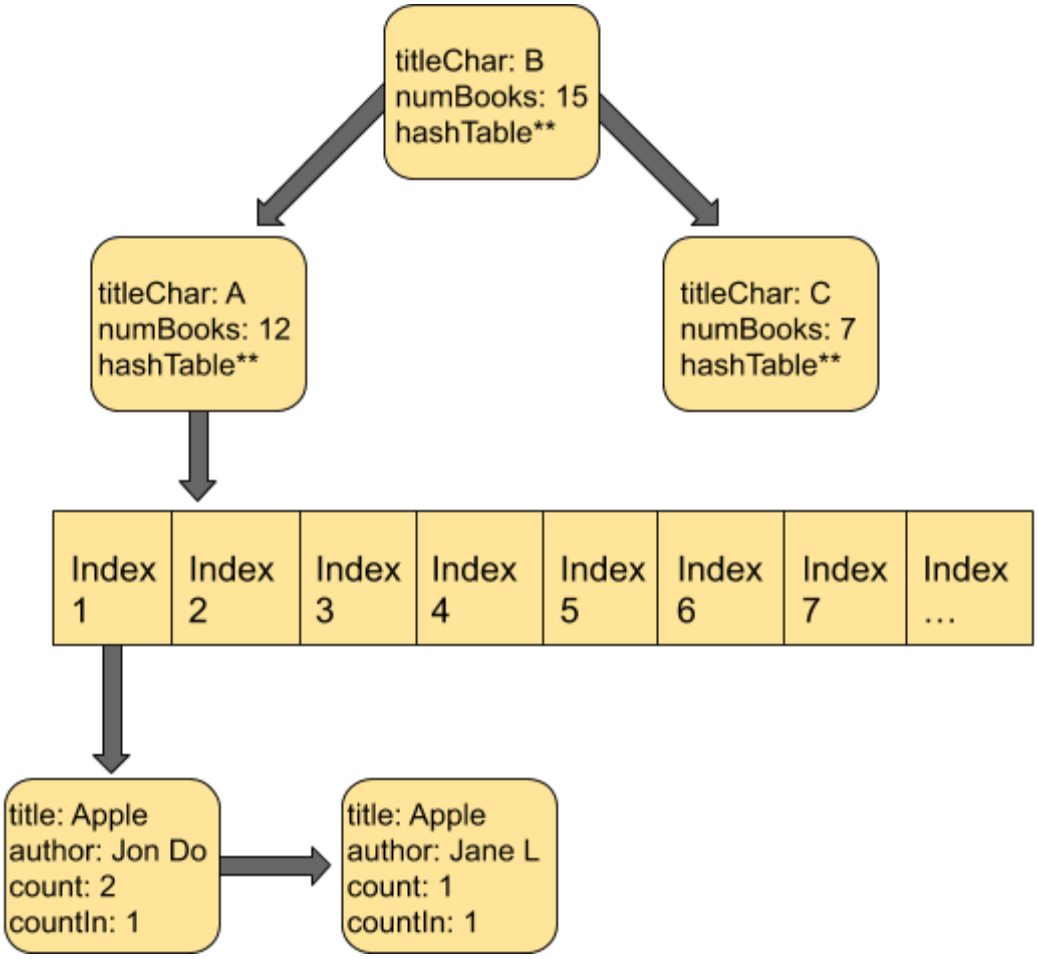
Final Project Report

Our project is a library inventory. Each book is stored in a bookNode struct which contains the title of the book, the author of the book, an integer count that stores how many copies of the book are in the inventory, an integer inCount that stores how many copies of the book are currently not checked out, and finally a pointer to the next bookNode.

The data structure used to store the library inventory is a Binary Search Tree with 23 treeNodes, one for each letter of the alphabet excluding F, Q and X. Each treeNode is a struct which contains a character variable titleChar that stores which letter of the alphabet that node is, a pointer to the left and right children, an integer value numBooks which is updated when a book is added or removed to the inventory with that particular letter, and finally a pointer to a hash table which stores each bookNode. To recap, our data structure is a Binary Search Tree where each node stores a hash table of books starting with that particular letter of the alphabet.

The reason we chose this particular set of data structures is because we knew that a balanced Binary Search Tree has a search time complexity of O(log n), and that searching for the first letter of the title of a book first then searching a smaller set of data (specifically a hash table containing only books with titles starting with the same first letter) would guarantee efficient time complexities for each of our functions.

Our main function acts as an interface for our program and allows for various actions. The main can be thought of as having two parts to it: the staff menu and the main menu. The main menu is the interface that the public would be able to access and use. The main menu contains our check in a book and check out a book functions, which allows a user to check a book in and out, along with a search by title and search by author options as well. The main menu also provides an option to take the user to the staff menu. If selected, the program will prompt the user to enter a password (by default, the password is ABCD). Once the password is correctly entered, the staff menu will open with more functionality for the user. There are add and remove book functions that allow the user to insert and remove books from the database. The user may also load in a .csv file that will automatically add books from the file. Each line of the file must be the title of the book and then the author. There is then a book query that will provide the user with detailed information about a book in the database. Finally, two functions that print checked out books and checked in books are available.


titleChar: B
numBooks: 15
hashTable**
titleChar: A
numBooks: 12
hashTable**
titleChar: C
numBooks: 7
hashTable**
Index 1
Index 2
Index 3
Index 4
Index 5
Index 6
Index 7
Index …
title: Apple
author: Jon Do
count: 2
countIn: 1
title: Apple
author: Jane L
count: 1
countIn: 1