Kaya Chou-Kudu
CS 232 Data Structures

Fall 2018 Final Project

How did you decide how to sort the books? How did you implement that in the CompareTo method?

I sorted my books first by genre (A-F), then by rating (highest to lowest). In the compareTo method, if book b has the same genre as book o, then it compares their ratings. Otherwise, it compares their genres.

Which sort did you pick, and why? Did you prioritize ease of coding, or efficiency of the algorithm?

I wrote two sort functions, a bubble sort and a selection sort. I left the bubble sort in my code, but the only one that I implement is the selection sort. I chose to ultimately implement selection sort because it’s more efficient than bubble sort.

How did you handle keeping track of an ArrayList for each genre, and what method(s) did you need to write to make sure that the ArrayList of Books was kept up-to-date and sorted?

Whenever a book was added to the library, the addBook() method checked if the hashtable contained that book’s genre name as a key. If it did, it added that book to the list (value) connected to that genre name (key). If not, it created a new genre key in the hashtable with a list as its value, and added the book to that list. Each time a book was added to a list, the list was sorted, and then put back into the hashtable in order to keep the lists sorted.

Kaya Chou-Kudu
CS 232 Data Structures

Describe how you implemented the Priority Queue, the code you used in the driver to test the edge cases for the Priority Queue, and what function(s) you wrote yourself.

I tried to implement an array-based priority queue with an add method, but it didn’t sort my books in the correct order. I tried really hard to figure out why it wasn’t working but I couldn’t figure it out even after debugging and testing it.