

Question Problem: Online Bookstore System (Text Based – No graphics Required)

Imagine you are tasked with designing the backend of an online bookstore system. Different data structures can be employed to handle various aspects of this system. Here's a problem description:

Design the data structures for an online bookstore system that allows customers to browse books, add them to their cart, and make purchases. Consider the following functionalities:

1. Book Catalog:

- ✓ Store information about each book, including its title, author, genre, price, and availability.
- ✓ Efficiently retrieve details of a specific book.

2. User Accounts:

- ✓ Store information about customers, including their username, password, email, and purchase history.
- ✓ Ensure efficient retrieval of user details.

3. Shopping Cart:

- ✓ Implement a data structure to represent the shopping cart for each user.
- ✓ Allow users to add and remove books from their cart.

4. Order History:

- ✓ Maintain a record of users' order history, including details such as order date, total price, and purchased items.

5. Recommendation System:

- ✓ Implement a system to recommend books to users based on their purchase history. No machine learning needed (not our focus here)

6. Inventory Management:

- ✓ Keep track of the inventory of each book and update it when a purchase is made.

7. Search and Filtering:

- ✓ Enable users to search for books based on various criteria, such as author, genre, or title.

Challenge:

Design the backend data structures to efficiently handle the above functionalities. Choose appropriate data structures for tasks like efficient book retrieval, user account management, and shopping cart operations. Consider the trade-offs between different data structures to optimize for performance and scalability. Document your design choices and explain why you selected each data structure for a specific task. **This problem allows you to explore the use of arrays, linked lists, hash maps, trees, and other data structures in the context of a real-world application. Space-Complexity analysis of each module (function) is also needed.**

Best of Luck!