**Exercise 6: Library Management System**

**Step 1: Understand Search Algorithms**

**Linear Search**

* **Description:** A simple search algorithm that checks each element in a list one by one until the desired element is found or the list ends.
* **Time Complexity:** O(n) for both best and worst cases.
* **Use Case:** Suitable for unsorted or small datasets.

**Binary Search**

* **Description:** A more efficient search algorithm that works on sorted lists. It repeatedly divides the search interval in half and compares the middle element with the target value.
* **Time Complexity:** O(log n) for both best and worst cases.
* **Use Case:** Suitable for large, sorted datasets.

**Time Complexity Comparison**

* **Linear Search:** O(n) - Iterates through the entire list until the element is found or the end is reached.
* **Binary Search:** O(log n) - Divides the search interval in half with each step, significantly reducing the number of comparisons.

**When to Use Each Algorithm**

* **Linear Search:**
  + Use for unsorted or small datasets where the overhead of sorting does not justify the performance gain.
  + Simple to implement and does not require the data to be sorted.
* **Binary Search:**
  + Use for large, sorted datasets where the performance improvement of O(log n) is significant.
  + Requires the data to be sorted, which may involve an additional O(n log n) sorting step if not already sorted.