**Exercise 6: Library Management System**

**1. Understand Search Algorithms**

**Linear Search**

* **Description**: Linear search is a simple search algorithm that checks each element in the list sequentially until the desired element is found or the list ends.
* **Time Complexity**: O(n)
  + **Best Case**: O(1) (if the desired element is the first element)
  + **Average Case**: O(n/2) ~ O(n)
  + **Worst Case**: O(n) (if the desired element is the last element or not in the list)

**Binary Search**

* **Description**: Binary search is a more efficient search algorithm that works on sorted lists. It repeatedly divides the list in half, comparing the target value to the middle element, and discards the half where the target cannot be.
* **Time Complexity**: O(log n)
  + **Best Case**: O(1) (if the desired element is the middle element)
  + **Average Case**: O(log n)
  + **Worst Case**: O(log n)

**4. Analysis**

**Time Complexity of Search Algorithms**

* **Linear Search**: O(n)
  + Best for unsorted or small datasets.
* **Binary Search**: O(log n)
  + Best for large, sorted datasets.

**When to Use Each Algorithm**

* **Linear Search**: Use when the dataset is small or unsorted. It's simple to implement and doesn't require the list to be sorted.
* **Binary Search**: Use when the dataset is large and sorted. It provides a significant performance improvement over linear search for large datasets.