**Exercise 6: Library Management System**

**1. Understand Search Algorithms:**

***o Explain linear search and binary search algorithms.***

***Linear Search:***

* **Time Complexity:** O(n)
* **Description:** Linear search examines each element in the list one by one until the desired element is found or the end of the list is reached. It's simple and works on unsorted data but can be slow for large datasets.

***Binary Search:***

* **Time Complexity:** O(log n)
* **Description:** Binary search repeatedly divides the search interval in half. It requires the data to be sorted. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half, or vice versa. It is much faster than linear search for large datasets due to its logarithmic time complexity.

**4. Analysis:**

***o Compare the time complexity of linear and binary search.***

***Linear Search:***

Best Case: 𝑂(1) — This occurs if the target value is the first element in the list.

Average Case: 𝑂(𝑛) — On average, the search will need to examine half of the elements in the list.

Worst Case: 𝑂(𝑛) — This occurs if the target value is at the end of the list or not present at all, requiring the search to check every element.

***Binary Search:***

Best Case: 𝑂(1) — This occurs if the target value is the middle element of the list.

Average Case: 𝑂(log𝑛) — On average, binary search will require log𝑛 comparisons to find the target value or determine that it is not in the list.

Worst Case: 𝑂(log𝑛) — This occurs when the target value is either present in the list or not, requiring log𝑛 comparisons.

***o Discuss when to use each algorithm based on the data set size and order.***

* **Linear Search:** Use when the dataset is small or unsorted.
* **Binary Search:** Use when the dataset is large and sorted.