**Exercise 6: Library Management System**

Scenario:

You are developing a library management system where users can search for books by title or author.

1. Explain linear search and binary search algorithms.

Linear Search is a straightforward algorithm used to find a target value within a list. It works by examining each element in the list sequentially until it either finds the target value or reaches the end of the list. If the target value matches the current element, the algorithm returns the index or the element itself. If not, it proceeds to the next element. This process continues until the target is found or the end of the list is reached.

Binary Search is a more efficient algorithm, but it requires the list to be sorted beforehand. It works by repeatedly dividing the search interval in half. The algorithm starts with the entire sorted list and determines the middle element. If the middle element matches the target value, the search is complete. If the target value is less than the middle element, the search continues in the left half of the list. Conversely, if the target is greater, the search continues in the right half. This process of halving the search interval continues until the target is found or the interval becomes empty.

Analysis:

o Compare the time complexity of linear and binary search.

**Time Complexity**

1. **Linear Search**
   * **Best Case:** O(1) — The element is found at the first position.
   * **Average Case:** O(n) — The element could be anywhere in the list, so on average, you might have to search through half of the elements.
   * **Worst Case:** O(n) — The element is at the last position or not in the list at all.
2. **Binary Search**
   * **Best Case:** O(1) — The middle element is the target.
   * **Average Case:** O(logn) — Each comparison halves the search space, so the number of comparisons grows logarithmically with the number of elements.
   * **Worst Case:** O(logn) — In the worst case, the element is found after searching through the entire reduced search space.

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

1. **Linear Search**
   * **Data Set Size:** Use linear search for small data sets where the overhead of sorting (for binary search) is not worth it.
   * **Order:** Linear search does not require the data to be sorted. It works with both sorted and unsorted lists.
   * **Advantages:** Simple to implement and does not require sorted data. Useful for data sets that change frequently, as sorting for binary search may be less practical.
2. **Binary Search**
   * **Data Set Size:** Ideal for large data sets where the efficiency of O(logn) makes a significant difference. The larger the data set, the more beneficial binary search becomes.
   * **Order:** Requires the data to be sorted. If the data is not sorted, you'll need to sort it first, which takes O(nlogn) time, possibly making binary search less efficient in such cases.
   * **Advantages:** Much faster than linear search for large sorted lists due to its logarithmic time complexity. Once data is sorted, binary search is very efficient for repeated searches.