**Exercise 6: Library Management System**

**Scenario:**

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

**Steps:**

1. **Understand Search Algorithms:**
   * **Explain linear search and binary search algorithms**.

**Linear Search:**

* Linear search checks each element of a list sequentially until the target element is found or the list ends.
* **Algorithm**:
  1. Start from the beginning of the list.
  2. Compare each element with the target value.
  3. If a match is found, return the index of the element.
  4. If the end of the list is reached without finding a match, return "not found".

**Binary Search:**

* Binary search works on sorted lists by repeatedly dividing the search interval in half. It requires the list to be sorted.
* **Algorithm**:
  1. Start with the entire sorted list.
  2. Compare the target value with the middle element of the list.
  3. If the target value equals the middle element, return the index.
  4. If the target value is less than the middle element, narrow the search to the left half of the list.
  5. If the target value is greater than the middle element, narrow the search to the right half of the list.
  6. Repeat steps 2-5 until the target is found or the search interval is empty.

1. **Analysis:**
   * **Compare the time complexity of linear and binary search.**

**Linear Search**:

* **Worst-case Time Complexity**: O(n)
* **Best-case Time Complexity**: O(1) (when the target is the first element)

**Binary Search**:

1. **Worst-case Time Complexity**: O(log n)
2. **Best-case Time Complexity**: O(1) (when the target is the middle element)
   * **Discuss when to use each algorithm based on the data set size and order**.

**Linear Search**:

* **Use Case**: Suitable for small or unsorted datasets where sorting is not feasible or necessary. Simple and does not require any prior arrangement of data.
* **Limitations**: Inefficient for large datasets due to its linear time complexity.

**Binary Search**:

* **Use Case**: Suitable for large datasets that are already sorted or can be sorted. More efficient than linear search for large sorted datasets due to its logarithmic time complexity.
* **Limitations**: Requires data to be sorted. The overhead of sorting can be significant if the data is not already sorted.