**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 is a straightforward algorithm that checks each element of the array sequentially until the target element is found or the end of the array is reached.

* **Best Case**: O(1) (when the target is the first element)
* **Average Case**: O(n)
* **Worst Case**: O(n)

**Binary Search**

Binary search is a more efficient algorithm for finding an element in a sorted array. It works by repeatedly dividing the search interval in half. If the value of the target is less than the middle element, the search continues on the left half, otherwise, it continues on the right half.

* **Best Case**: O(1) (when the target is the middle element)
* **Average Case**: O(log n)
* **Worst Case**: O(log n)

1. **Setup:**
   * Create a class **Book** with attributes like **bookId**, **title**, and **author**.
2. **Implementation:**
   * Implement linear search to find books by title.
   * Implement binary search to find books by title (assuming the list is sorted).
3. **Analysis:**
   * Compare the time complexity of linear and binary search.

**Time Complexity Comparison**

* **Linear Search**:
  + Best Case: O(1) (if the target is the first element)
  + Average Case: O(n)
  + Worst Case: O(n)
* **Binary Search**:
  + Best Case: O(1) (if the target is the middle element)
  + Average Case: O(log n)
  + Worst Case: O(log n)
  + Discuss when to use each algorithm based on the data set size and order.

**Linear Search**: Use linear search when:

* The dataset is small, as the overhead of sorting the data for binary search might not be justified.
* The dataset is unsorted, and there are no plans to sort it.
* Simplicity is preferred, and performance is not a critical concern.

**Binary Search**: Use binary search when:

* The dataset is large and searching operations are frequent.
* The dataset is sorted or can be sorted efficiently.
* Fast search performance is critical.