**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.
2. **Setup:**
   * Create a class **Book** with attributes like **bookId**, **title**, and **author**.
3. **Implementation:**
   * Implement linear search to find books by title.
   * Implement binary search to find books by title (assuming the list is sorted).
4. **Analysis:**
   * Compare the time complexity of linear and binary search.
   * Discuss when to use each algorithm based on the data set size and order.

**Explanation**

**Step 1: Understand Search Algorithms**

**Linear Search:**

* **Explanation:** Linear search is a straightforward algorithm that checks each element in a list one by one until the desired element is found or the list ends.
* **Time Complexity:**
  + **Best Case:** O(1) - The element is at the beginning of the list.
  + **Average Case:** O(n) - The element is somewhere in the middle of the list.
  + **Worst Case:** O(n) - The element is at the end of the list or not present.

**Binary Search:**

* **Explanation:** Binary search is a more efficient algorithm that repeatedly divides a sorted list in half to locate an element. It compares the target value to the middle element of the list, reducing the search range by half each time.
* **Time Complexity:**
  + **Best Case:** O(1) - The element is in the middle of the list.
  + **Average Case:** O(log n) - The element is found after a few halving steps.
  + **Worst Case:** O(log n) - The element is found after log n steps.

**Step 4: Analysis**

**Time Complexity Comparison:**

* **Linear Search:**
  + **Best Case:** O(1)
  + **Average Case:** O(n)
  + **Worst Case:** O(n)
* **Binary Search:**
  + **Best Case:** O(1)
  + **Average Case:** O(log n)
  + **Worst Case:** O(log n)

**When to Use Each Algorithm:**

* **Linear Search:**
  + **Use Case:** Small datasets or unsorted data.
  + **Advantage:** Simple and does not require the data to be sorted.
  + **Disadvantage:** Inefficient for large datasets as it checks each element sequentially.
* **Binary Search:**
  + **Use Case:** Large datasets where the data is sorted.
  + **Advantage:** Much faster than linear search for large datasets due to its logarithmic time complexity.
  + **Disadvantage:** Requires the data to be sorted, which may add additional overhead for maintaining the sorted order.