**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 in a list until the target element is found or the list ends.
* Time complexity: O(n), where n is the number of elements in the list.
* It works well for small or unsorted lists.

**Binary Search:**

* Binary search is a more efficient algorithm that works on sorted lists.
* It repeatedly divides the search interval in half, comparing the target value to the middle element of the interval.
* Time complexity: O(log n), where n is the number of elements in the list.
* It is much faster than linear search for large, sorted lists.

1. **Setup:**

A class **Book** is created with attributes like **bookId**, **title**, and **author**.

1. **Implementation:**

The implementation code is provided in the wordpad.

1. **Analysis:**
   1. **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) (if the target is the last element or not present)
* **Binary Search:**
  + Best Case: O(1) (if the target is the middle element)
  + Average Case: O(log n)
  + Worst Case: O(log n) (if the target is not found)
  1. **Discuss when to use each algorithm based on the data set size and order.**

**When to Use Each Algorithm:**

* **Linear Search:**
  + Suitable for small or unsorted lists.
  + Simple to implement.
  + No need for the list to be sorted.
* **Binary Search:**
  + Suitable for large, sorted lists.
  + More efficient than linear search for large datasets.
  + Requires the list to be sorted beforehand, which adds preprocessing time.