**Exercise 6: Library Management System**

**Scenario:**

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

1. **Understand Search Algorithms:**

**Linear Search**

**Algorithm Explanation:**

* Linear search, also known as sequential search, involves checking each element of the list one by one until the desired element is found or the list ends.
* It starts at the first element and moves sequentially through the list, comparing each element with the target value.
* If a match is found, the index of the element is returned; otherwise, the search continues until the end of the list.

**Binary Search**

**Algorithm Explanation:**

* Binary search is a more efficient algorithm that requires the list to be sorted.
* It works by repeatedly dividing the search interval in half.
* It starts by comparing the target value to the middle element of the list.
* If the target value is equal to the middle element, the search is complete.
* If the target value is less than the middle element, the search continues on the left half of the list.
* If the target value is greater than the middle element, the search continues on the right half of the list.
* This process repeats until the target value is found or the interval is empty.

**4. Analysis:**

**Time Complexity Comparison**

* **Linear Search:**
  + Best Case: O(1) (if the target is at the first position)
  + Worst Case: O(n) (if the target is at the last position or not present)
  + Average Case: O(n)
* **Binary Search:**
  + Best Case: O(1) (if the target is at the middle position)
  + Worst Case: O(logn)
  + Average Case: O(logn)

**When to Use Each Algorithm**

* **Linear Search:**
  + **Unsorted Data:** Linear search can be used on unsorted data because it does not rely on any order within the list.
  + **Small Data Sets:** For small lists, the simplicity of linear search can make it faster due to the lower constant factors in its time complexity.
  + **Single or Few Searches:** When performing only a single or a few searches, linear search might be preferred because the overhead of sorting the list for binary search might not be justified.
* **Binary Search:**
  + **Sorted Data:** Binary search can only be used on sorted lists, as it relies on the ability to split the search interval.
  + **Large Data Sets:** For large lists, the O(logn) time complexity of binary search makes it significantly faster than linear search.
  + **Frequent Searches:** If multiple searches are to be performed, the initial cost of sorting the list (if it is not already sorted) can be outweighed by the benefits of faster search times.