**Exercise 6: Library Management System**

**Step 1: Understand Search Algorithms**

**Linear Search:**

* **How it works:** Linear search goes through each element in a list one by one, starting from the first element, until it finds the target or reaches the end of the list.
* **Time Complexity:** O(n)O(n)O(n), where nnn is the number of elements in the list.
* **Use Case:** Useful for small or unsorted datasets.

**Binary Search:**

* **How it works:** Binary search works on sorted lists. It repeatedly divides the list in half and compares the middle element with the target. If the middle element is the target, the search ends. If the target is smaller, the search continues in the left half; if larger, in the right half.
* **Time Complexity:** O(log⁡n)O(\log n)O(logn), where nnn is the number of elements in the list.
* **Use Case:** Best for large, sorted datasets.

**Step 4: Analysis**

**Time Complexity Comparison:**

* **Linear Search:**
  + **Time Complexity:** O(n)O(n)O(n)
  + **Advantage:** Simple to implement; works on unsorted data.
  + **Disadvantage:** Slow for large datasets.
* **Binary Search:**
  + **Time Complexity:** O(log⁡n)O(\log n)O(logn)
  + **Advantage:** Much faster for large, sorted datasets.
  + **Disadvantage:** Requires the data to be sorted.

**When to Use Each Algorithm:**

* **Linear Search:** Use when the dataset is small or unsorted.
* **Binary Search:** Use when the dataset is large and sorted. It is much more efficient in terms of time complexity but requires the list to be sorted first.