**Exercise 6: Library Management System**

1. **Understanding Search Algorithms:**
   * **Q) Explain linear search and binary search algorithms.**
     + **Ans)**
       - **Linear Search**: This algorithm searches each element of the list sequentially to find the required element, continuing until the end of the list is reached.
         * **Time Complexity**: O(n), where n is the total number of elements in the list.
       - **Binary Search**: This efficient algorithm requires the list to be sorted. It works by repeatedly dividing the list in half and comparing the target value with the middle element until the target value is found.
         * **Time Complexity**: O(log n), where n is the number of elements in the list.
2. **Analysis:**
   * **Q) Compare the time complexity of linear and binary search.**
     + **Ans)**
       - **Linear Search**:
         * **Best Case**: O(1), if the element is the first one.
         * **Average Case**: O(n).
         * **Worst Case**: O(n).
       - **Binary Search**:
         * **Best Case**: O(1), if the element is the middle one.
         * **Average Case**: O(log n).
         * **Worst Case**: O(log n).
   * **Q) Discuss when to use each algorithm based on the data set size and order.**
     + **Ans)**
       - **Linear Search**: Suitable for small datasets or when the data is not sorted. It is easier to implement and does not require the data to be sorted.
       - **Binary Search**: Ideal for large datasets where the data is already sorted. It is much more efficient for larger datasets due to its logarithmic time complexity. Although it requires the data to be sorted, which adds some overhead, this can be minimized if the data is already sorted or changes infrequently.