Library Management System

- Linear Search:

Description: Linear search sequentially checks each element of the list until it finds the target element or reaches the end of the list.

Time Complexity:

- Best Case: O (1) (element is at the first position)
- Average Case: O(n)
- Worst Case: O(n)

- Binary Search:

Description: Binary search is used on a sorted array. It repeatedly divides the search interval in half. If the value of the target element is less than the middle element, the search continues on the left subarray; otherwise, it continues on the right subarray.

Time Complexity:

- Best Case: O (1)
- Average Case: O (log n)
- Worst Case: O (log n)

- 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:

- Suitable for small datasets where the overhead of sorting is not justified.
- Effective when the dataset is not sorted and needs to be searched a few times.
- Useful for datasets that are frequently updated, where maintaining a sorted order is not practical.

Binary Search:

- Suitable for large datasets where fast search times are crucial.
- Requires the dataset to be sorted, which can add an initial overhead if the dataset changes frequently.
- Ideal for datasets that do not change often and where search operations are frequent.