6. Library Management System

Linear Search and Binary Search

1. Linear Search
Description: Linear Search is the simplest search algorithm. It works by checking each element in a list sequentially until the desired element is found or the list ends.
Process:

- Start from the first element of the list.
- Compare each element with the target value.
- If a match is found, return the index of the element.
- If no match is found after checking all elements, indicate that the element is not present.

2.Binary Search
Description: Binary Search is a more efficient search algorithm that works on sorted arrays or lists. It divides the search interval in half repeatedly until the target value is found or the search space is exhausted.
Process:

- Start with the entire sorted list.
- Compare the target value to the middle element of the list.
- If the target is equal to the middle element, the search is complete.
- If the target is less than the middle element, repeat the search on the left half of the list.
- If the target is greater than the middle element, repeat the search on the right half.
- Continue this process until the target is found or the search interval is empty.

Analysis

Analysis of Time Complexity

1. **Linear Search**:
    - **Best Case**: O(1) - The target is the first element.
    - **Average Case**: O(n) - The target is found midway through the list.
    - **Worst Case**: O(n) - The target is the last element or not present.
2. **Binary Search**:
    - **Best Case**: O(1) - The target is the middle element.
    - **Average Case**: O(logn) - Each comparison halves the search space.
    - **Worst Case**: O(logn) - The search space is reduced to zero after several comparisons.

Comparison of Linear and Binary Search

- **Efficiency**:
    - Linear Search has a linear time complexity, making it inefficient for large datasets.

    - Binary Search is significantly faster for large, sorted datasets due to its logarithmic time complexity.
- **Data Requirements**:
    - Linear Search can be used on both sorted and unsorted data, making it versatile.
    - Binary Search requires the data to be sorted beforehand.

When to Use Each Algorithm

1. **Linear Search**:
    - Use when the dataset is small or when the data is unsorted.
    - Suitable for situations where simplicity is preferred over efficiency.
    - Best for cases where elements are frequently added or removed, making sorting impractical.
2. **Binary Search**:
    - Use when working with large, sorted datasets.
    - Ideal for applications where quick lookups are essential.
    - Best when the dataset is static or changes infrequently, allowing for efficient sorting.