**LIBRARY MANAGEMENT**

1.Understand the problem

* **Explain linear search and binary search algorithms.**

**Linear Search**:Linear search is a simple search algorithm that checks each element of a list one by one until it finds the desired element. It has a time complexity of O(n), where n is the number of elements in the list.

**Binary Search:**  Binary search is a more efficient search algorithm that works by dividing the list in half and repeatedly searching for the desired element in one of the two halves. It has a time complexity of O(log n), where n is the number of elements in the list. However, binary search requires the list to be sorted in ascending or descending order.

4.Analysis

* **Compare the time complexity of linear and binary search.**
  + **Linear Search:**
    - Best Case: O(1) (if the target element is the first one)
    - Average Case: O(n)
    - Worst Case: O(n)
  + **Binary Search:**
    - Best Case: O(1) (if the target element is the middle one)
    - Average Case: O(log n)
    - Worst Case: O(log n)
* **Discuss when to use each algorithm based on the dataset size and order.**

**Linear Search:**

* Best for small or unsorted datasets.
* Easy to implement and does not require the data to be pre-sorted.
* Suitable when the cost of sorting the data exceeds the benefits of faster search times.

**Binary Search:**

* Ideal for large, sorted datasets.
* Requires the data to be sorted beforehand, which may involve an initial cost but allows for significantly faster searches afterwards.
* Most effective when performing multiple searches on a static dataset, as the initial sorting cost is spread over many search operations.