**Library Management System**

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

1. **Linear Search**:

Linear search is a simple search algorithm that checks each element in the list one by one until the desired element is found or the list ends. It is suitable for small or unsorted datasets.

**Time Complexity**: O(n) where n is the no. of elements in the list.

1. **Binary Search**:

Binary search is an efficient algorithm for finding an element in a sorted list. It repeatedly divides the list into halves, comparing the target value to the middle element, and narrowing down the search to one half of the list. It is suitable for large, sorted datasets.

**Time Complexity**: O(log n) where n is the no. of elements in the list.

**Time Complexity Analysis:**

* 1. **Compare the time complexity of linear and binary search.**

L**inear Search:**

Time complexity is O(n) in the worst-case. Means the search time grows linearly with the number of elements in the data set (n). In the worst case, the search might need to iterate through all elements before finding a match.

**Binary Search:**

Time complexity is O(log n) in the worst-case. Means the search time grows logarithmically with the data set size. With each comparison, the search space is divided in half, leading to a faster search for larger datasets.

* 1. **Discuss when to use each algorithm based on the data set size and order.**

The choice between linear and binary search depends on:

* **Data Set Order:**  **Binary search** requires the **data set to be sorted** in a specific order (ascending or descending based on the comparison function). **Linear search** works on **both sorted and unsorted data sets.**
* **Data Set Size:** For very small datasets, the difference between **linear** and **binary search** might be negligible. However, as the data set size grows, **binary search** becomes **significantly faster** due to **its logarithmic time complexity**.