**Exercise 6: Library Management System**

Explain linear search and binary search algorithms.

Linear Search:

Linear search iterates through each element in the list until the target element is found or the end of the list is reached.

Time Complexity:

* Best Case: O(1) - When the target element is the first element.
* Worst Case: O(n) - When the target element is the last element or not present.

Binary Search:

Binary search works on sorted arrays by repeatedly dividing the search interval in half. If the target value is less than the middle element, the search continues in the lower half, otherwise in the upper half.

Time Complexity:

* Best Case: O(1) - When the target element is the middle element.
* Worst Case: O(log n) - When the search interval is halved each time.

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

Time Complexity:

* Linear Search:
  + Best Case: O(1) - When the target element is the first element.
  + Average Case: O(n) - On average, half of the elements are checked.
  + Worst Case: O(n) - When the target element is the last element or not present.
* Binary Search:
  + Best Case: O(1) - When the target element is the middle element.
  + Average Case: O(log n) - When the search interval is halved each time.
  + Worst Case: O(log n) - When the search interval is halved each time.

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

Linear Search: Use when the dataset is small or unsorted. It's straightforward and doesn't require pre-sorting the data.

Binary Search: Use when the dataset is large and sorted. It significantly reduces the search time compared to linear search.