**Algorithm\_Data Structures**

**Exercise 6: Library Management System**

**Search Algorithms:**

This project helps us build a book search system using basic search algorithms, Linear Search and Binary Search. These algorithms are used to find a book in the library by matching its bookId. Linear Search can work with any list, even if it’s not sorted. Binary Search is faster, but it needs the list to be sorted by bookId first.

1) Linear Search: Linear Search checks each book’s bookId one by one, starting from the first to the last. It’s very simple and doesn’t need the data to be sorted. It stops as soon as the matching bookId is found. Best suited for small or unsorted book lists.

2) Binary Search: Binary Search is faster but only works if the book list is sorted by bookId. It starts in the middle of the list and decides whether to look in the left or right half. It cuts the search space in half at each step.

**Time Complexity for each search method:**

**Linear Search:** This function searches for a matching bookId by going through the array from beginning to end.

Time complexity is O(n) because it may need to check all books in the worst case.

**Binary Search**: This function searches for a matching bookId in a sorted array by dividing the list repeatedly.

Time complexity is O(log n) as the search range is halved each time.

**When to Use Linear Search:**

Linear Search is best used when the number of books in the system is small. In such cases, the time taken to search through the list is minimal, and there’s no need for complex logic. It is also useful when the list of books is not sorted by bookId. Since Linear Search does not rely on any particular order, it works well even when the data is unordered. Another important case is when books are added or removed frequently. In such situations, keeping the list sorted becomes difficult and time-consuming. Linear Search handles this better because it doesn’t require sorting after every update. Although Linear Search is slower for large lists, it is easier to implement and manage in dynamic systems with fewer records.

**When to Use Binary Search:**

Binary Search is ideal when the list of books is large and already sorted by bookId. It is much faster than Linear Search because it cuts the search range in half during each step, significantly reducing the number of comparisons. If the data set is stable and doesn’t change often, Binary Search can give very quick results, making it perfect for performance-critical applications like library catalog systems with many books. However, Binary Search requires the data to be sorted. If books are added or removed frequently, the list must be re-sorted before each search, which adds extra processing. So, Binary Search is most effective when the data is mostly static and search speed is a priority.