

Exercise 6: Library Management System

Analysis:

- Compare the time complexity of linear and binary search.
- Discuss when to use each algorithm based on the data set size and order.

Time Complexity:

Search Algorithm	Best Case	Average Case	Worst Case
Linear search	$O(1)$	$O(n)$	$O(n)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$

Explanation:

- Linear Search scans each record one by one.
- Binary Search divides the search range in half repeatedly—but requires the data (e.g., book list) to be sorted.

When to Use Each Algorithm

Linear Search

Condition	Reason
Data is unsorted	Binary search can't be used without sorting first
Data is small	Overhead of sorting or indexing is unnecessary
One time or rare search	Simple and doesn't require preprocessing

Binary Search:

Condition	Reason
Dataset is large and sorted	Much faster search ($O(\log n)$) for thousands of books
Frequent searching needed	Sorting once ($O(n \log n)$) + binary search many times is efficient
Data rarely changes	Maintaining sorted order is easier when insertions/deletions are rare

Conclusion

- Binary Search is ideal for large, sorted datasets and frequent search operations in a library system.
- Linear Search is better for small or unsorted datasets, or when sorting is not practical.