

Analysis

Linear Search

Linear search involves checking each book in the list one by one until the desired book is found or the end of the list is reached. It does **not require the list to be sorted** and is simple to implement.

Time Complexity: $O(n)$, where n is the number of books.

Advantages:

- Works on unsorted data
- Simple and easy to implement

Limitations:

- Inefficient for large datasets, as it may need to scan every item
-

Binary Search

Binary search is significantly faster but **requires the list to be sorted** (e.g., alphabetically by title). It works by repeatedly dividing the search interval in half and narrowing down the location of the target book based on comparisons.

Time Complexity: $O(\log n)$

Advantages:

- Very efficient for large, sorted datasets
- Requires fewer comparisons (e.g., only about 20 steps for 1 million books)

Limitations:

- Requires pre-sorting
 - Slightly more complex to implement
-

Summary Comparison

Search Type	Requires Sorted	Data Time Complexity	Best Used For
Linear Search	No	$O(n)$	Small or unsorted collections
Binary Search	Yes	$O(\log n)$	Large and sorted collections

Conclusion

Linear search is most appropriate for small or frequently updated (unsorted) lists due to its simplicity. In contrast, binary search is highly efficient for large and sorted collections, making it suitable for optimized systems that prioritize search speed.