Step 1: Understand Search Algorithms

Linear Search:

Algorithm:

Start from the beginning of the list and check each element sequentially until the target element is found or the end of the list is reached.

Time Complexity:

Worst Case: O(n) – In the worst-case scenario, you might need to examine every element in the list.

Best Case: O(1) – If the target element is the first one, the search completes immediately.

Advantages:

Simple to implement.

Works on unsorted data.

Disadvantages:

Inefficient for large data sets compared to binary search.

Binary Search:

Algorithm:

Divide the sorted list into halves. Compare the target value with the middle element. If the target value is less, search the left half; otherwise, search the right half. Repeat this process until the target is found or the search interval is empty.

Time Complexity:

Worst Case: O(log n) – Each comparison effectively halves the search space.

Best Case: O(1) – If the target element is the middle element, the search completes immediately.

Advantages:

Much faster than linear search for large datasets.

Requires sorted data.

Disadvantages:

Needs sorting of the list if it is not already sorted.

4. Analysis:

o Compare the time complexity of linear and binary search.

Linear Search:

Time Complexity: O(n) – The search may require checking each book in the list.

Space Complexity: O(1) – Only a few extra variables are used.

Binary Search:

Time Complexity: O(log n) – Requires a logarithmic number of comparisons.

Space Complexity: O(1) – Only a few extra variables are used, assuming the sorting is done in-place.

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

Linear Search:

Use When: The data is unsorted or the dataset is small.

Advantage: Does not require the data to be sorted.

Disadvantage: Less efficient for large datasets.

Binary Search:

Use When: The data is sorted or can be sorted beforehand.

Advantage: Much faster than linear search for large datasets.

Disadvantage: Requires sorting the data, which adds overhead if the data is not already sorted.

For the library management system, if the dataset of books is small or unsorted, linear search might be sufficient. However, for larger datasets or when the list of

books can be kept sorted, binary search offers a significant performance improvement. Sorting the list may be an additional overhead, but the benefits of faster

search operations for large datasets usually outweigh the cost.