1. Understand Search Algorithms

Q. Explain linear search and binary search algorithms?

=>Linear Search

Description: Linear search is a straightforward algorithm that checks each element in a list sequentially until the desired element is found or the list ends.

Time Complexity:

Best Case: O(1) (if the element is the first one)

Average Case: O(n)

Worst Case: O(n)

Binary Search

Description: Binary search is a more efficient algorithm that works on sorted lists. It repeatedly divides the search interval in half, comparing the target value to the middle element of the interval and adjusting the search range based on the comparison.

Time Complexity:

Best Case: O(1) (if the element is the middle one)

Average Case: O(log n)

Worst Case: O(log n)

4. Analysis

Q. Compare the time complexity of linear and binary search ?

=>Linear Search:

Best Case: O(1) (if the element is the first one)

Average Case: O(n)

Worst Case: O(n)

Binary Search:

Best Case: O(1) (if the element is the middle one)

Average Case: O(log n)

Worst Case: O(log n)

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

=>Linear Search:

1.Suitable for small or unsorted datasets.

2.Simple to implement and does not require the data to be sorted.

3.Useful when the cost of sorting the data outweighs the benefits of faster searches.

Binary Search:

1.Suitable for large, sorted datasets.

2.Requires initial sorting of the data, which can be time-consuming but allows for much faster search times afterwards.

3.Ideal when multiple searches are needed on a static dataset, as the overhead of sorting is amortized over many search operations.