1. Explain linear search and binary search algorithms.

Linear Search:

Sequentially checks each element in a list until the target element is found or the end of the list is reached.

Time Complexity: O(n)

Use Case: Suitable for unsorted or small lists where the overhead of sorting is not justified.

Binary Search:

Efficiently finds a target element in a sorted list by repeatedly dividing the search interval in half.

Time Complexity: O(log n)

Use Case: Ideal for large, sorted lists due to its logarithmic time complexity.

1. Compare the time complexity of linear and binary search.

Linear Search: O(n)

Suitable for small or unsorted datasets where the overhead of sorting for binary search is not justified.

Binary Search: O(log n)

Efficient for large, sorted datasets but requires the list to be sorted.

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

Linear Search: Use for small or unsorted lists where sorting is impractical or unnecessary.

Binary Search: Use for large, sorted lists where the cost of sorting is offset by the efficiency of the search