Explain linear search and binary search algorithms.

Linear Search

Linear search is a simple algorithm that sequentially checks each element in a list until the target element is found or the end of the list is reached. It does not require the list to be sorted.

- **Time Complexity**: O(n) - In the worst case, the algorithm needs to check each element in the list.
- **Space Complexity**: O(1) - It requires a constant amount of additional space.

Binary Search

Binary search is a more efficient algorithm that works on sorted lists. It repeatedly divides the search interval in half. If the target value is less than the middle element, the search continues in the lower half; otherwise, it continues in the upper half.

- **Time Complexity**: O(log n) - The algorithm divides the search space in half at each step.
- **Space Complexity**: O(1) - It requires a constant amount of additional space.

Compare the time complexity of linear and binary search.

Linear Search: O(n) - The algorithm checks each element in the list sequentially.

Binary Search: O(log n) - The algorithm divides the search space in half at each step

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

- **Linear Search**: Use when the list is unsorted or small. It is simple to implement and does not require the list to be sorted.
- **Binary Search**: Use when the list is sorted and large. It is much more efficient for large datasets due to its logarithmic time complexity.