**1)Explain linear search and binary search algorithms.**

Linear Search:

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

Works on both sorted and unsorted lists.

Time complexity: O(n) where n is the number of elements.

Binary Search:

Works only on sorted lists.

Repeatedly divides the search interval in half.

Time complexity: O(log n) where n is the number of elements.

**2) Compare the time complexity of linear and binary search.**

Time Complexity Comparison:

Linear Search: O(n)

Pros: Works on unsorted lists, simple to implement

Cons: Slow for large datasets

Binary Search: O(log n)

Pros: Much faster for large datasets

Cons: Requires sorted list, more complex to implement.

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

Use Linear Search when:

The list is unsorted and sorting would be too costly.

The list is small (typically less than 100 elements).

You need to find all occurrences of an element, not just one.

The data structure doesn't support random access (e.g., linked lists).

Use Binary Search when:

The list is already sorted or can be sorted efficiently.

The list is large (typically more than 100 elements).

You only need to find one occurrence of an element.

The data structure supports random access (e.g., arrays, ArrayLists).