Q6. Understand Search Algorithms:

*a) Linear Search*:

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

Simple to implement but inefficient for large datasets.

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

*b) Binary Search*:

Works on sorted arrays by repeatedly dividing the search interval in half.

More efficient for large datasets but requires the data to be sorted.

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

Analysis:

Time complexity comparison:

* Linear Search: O(n)
* Binary Search: O(log n)

When to use each algorithm:

*Linear Search*:

* Use when the data is unsorted.
* Efficient for small datasets.
* When searching is infrequent and the cost of sorting is not justified.
* When elements are being frequently added or removed from the list.

*Binary Search*:

* Use when the data is sorted or can be sorted once and searched many times.
* Efficient for large datasets.
* When search operations are frequent and justify the initial sorting cost.
* When the dataset is relatively stable (not many additions or removals).