**1.**

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

**Ans :** Types of Linked Lists

* **Linear Search** : Checks each element in a list sequentially until the target element is found or the list ends. Simple to implement but inefficient for large lists.
* **Binary Search :** Efficient search algorithm that requires the list to be sorted. It repeatedly divides the search interval in half. More efficient than linear search for large, sorted lists.

**4.**

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

**Ans :**

* **Linear Search**:

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

**Average Case**: O(n) (on average, you check half of the elements)

**Worst Case**: O(n) (if the target is the last element or not present)

* **Binary Search**:

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

**Average Case**: O(log n) (divides the search space by half each step)

**Worst Case**: O(log n) (same as average case)

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

**Ans :**

* **Linear Search:** It is used in searching small datasets or in an unsorted list. The simplest approach, which does not involve any pre-sorting, is linear search.
* **Binary Search :** This only makes sense on a large, sorted data set. It is more efficient but does require that the data is sorted.