Que1. Explain linear search and binary search algorithms?

Ans: Linear Search:

* Algorithm: Linear search involves checking each element in the list sequentially until the target element is found or the end of the list is reached.
* Time Complexity: O(n)
* Best Case: O(1) (if the target element is the first element)
* Worst Case: O(n) (if the target element is not in the list or is the last element)

Binary Search:

* Algorithm: Binary search works on a sorted list by repeatedly dividing the search interval in half. If the target element is less than the middle element, it searches the left half; otherwise, it searches the right half.
* Time Complexity: O(log n)
* Best Case: O(1) (if the target element is the middle element)
* Worst Case: O(log n) (dividing the list in half each time)

Que2. Compare the time complexity of linear and binary search?

Ans: Linear Search:

* Time Complexity: O(n)
* Space Complexity: O(1)

Binary Search:

* Time Complexity: O(log n)
* Space Complexity: O(1) (iterative) or O(log n) (recursive)

Que3. Discuss when to use each algorithm based on the data set size and order?

Ans: You can use linear search when the list size is small and unsorted but use binary search if the size of the list is large because binary search is more efficient than linear search.