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

**Linear search** is a straightforward searching technique where each element in a list is checked one by one until the desired value is found or the list ends. It works on both sorted and unsorted data but can be slow for large lists since it may need to examine every item.

**Binary search**, on the other hand, is a faster method used only on sorted data. It works by repeatedly dividing the list in half, comparing the middle element with the target value, and narrowing down the search to either the left or right half. This method significantly reduces the number of comparisons, making it much more efficient for large datasets.

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

Linear search has a time complexity of O(n), where ‘n’ is the total number of elements in the dataset. This means that in the worst case scenario, the algorithm may need to scan through each item one by one until it either finds the target or reaches the end of the list. As a result, linear search can become inefficient as the size of the dataset grows, especially when the desired item is near the end or not present at all.

Binary search, in contrast, is much more efficient for large datasets, with a time complexity of O(log n). It works by repeatedly dividing the sorted list into halves and comparing the target value to the middle element. Based on the comparison, it eliminates one half of the list from consideration and continues the search in the other half. This process significantly reduces the number of steps needed to find the item. However, binary search only works if the dataset is already sorted, and maintaining this order can require extra work, especially if items are frequently added or removed.

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

Use linear search when the dataset is small, unsorted, or changes frequently, as it is simple to implement and doesn’t require sorting. It's also useful when the list is short and speed is not a major concern. On the other hand, binary search is ideal for large, sorted datasets where performance matters. Since it repeatedly divides the search range in half, it drastically reduces the number of comparisons, making it more efficient but it only works if the data remains sorted.