E-COMMERCE

Q1)Understand Asymptotic Notation:
Asymptotic notation is a way to describe the performance or complexity of an algorithm, which is crucial in analyzing and comparing algorithms. There are three main types of asymptotic notation: Big O, Omega, and Theta

Big O notation :
Big O notation is a measure of the upper bound of an algorithm's complexity, usually expressed as a function of the input size n. It's denoted as O(f(n)), where f(n) is the function that describes the algorithm's complexity.
Big O notation gives an estimate of the worst-case scenario, i.e., the maximum time an algorithm takes to complete. It is a mathematical notation that describes the complexity of an algorithm, which is the amount of time or space it requires as the input size grows.

Best-Case Scenario :
The best-case scenario occurs when the search element is found at the first position, and the algorithm terminates immediately. The time complexity for the best-case scenario is typically O(1), indicating constant time complexity.

Average-Case Scenario :
The average-case scenario assumes that the search element is randomly located in the data structure. The time complexity for the average-case scenario is usually O(n/2) or O(log n), depending on the search algorithm used.

Worst-Case Scenario :
The worst-case scenario occurs when the search element is not found in the data structure, and the algorithm must traverse the entire data structure. The time complexity for the worst-case scenario is typically O(n), indicating linear time complexity.

Linear Search:
Best Case: O(1)
Average Case: O(n)
Worst Case: O(n)

Binary Search:
Best case: O(1)
Average case: O(log n)
Worst case: O(log n)

Q4) Time Complexity Analysis:
Linear Search: O(n)
Binary Search:O(log n)

Comparison and Suitability:

Linear Search: Good for small datasets or unsorted data. It's simple but inefficient for large datasets due to O(n) time complexity.
Binary Search: Efficient for large datasets due to O(log n) time complexity but requires the data to be sorted. It is suitable for scenarios where search speed is critical and the dataset is large.

Comparision and Suitability for E-commerce Platform:
Linear search checks each element in the array until it finds the target element, resulting in a linear increase in time as the input size (n) increases. It is simple to implement but can be slow for large input sizes.
Binary search works by dividing the search space in half at each step, resulting in a logarithmic decrease in time as the input size (n) increases.
It is more efficient than linear search, especially for large input sizes.
Thus, based on the time complexity comparison, binary search is more suitable for the e-commerce platform. It also gives the users a faster search results as well as it ensures that the search functionality will remain efficient and scalable, even with a growing product catalog.