**1.**

* **Explain Big O notation and how it helps in analyzing algorithms.**

**Ans : Explanation :** Big O notation is a mathematical notation that describes the upper bound of the running time of an algorithm. It gives a high-level understanding of the efficiency of an algorithm, classifying its time complexity as it relates to the input size.

Big O notation is used to analyze the performance or to compare the performance of any algorithms on their growth rates as the input size increases.

* **Describe the best, average, and worst-case scenarios for search operations.**

**Ans :**

* **Best Case**: The searching scenario where the element is found at the start or at the first position are known as Best Case Scenario.
* **Average Case**: The searching scenario where the element is found at the middle of a list or an array are known as Average Case Scenario.
* **Worst Case**: The searching scenario where the element is found at the last position of the list or an array are known as Worst Case Scenario.

**4.**

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

**Ans :**

* **Linear Search**:

Best Case Time Complexity: O(1)

Worst Case Time Complexity: O(n)

Average Case Time Complexity: O(n)

Space Complexity: O(1)

* **Binary Search**:

Best Case Time Complexity: O(1)

Worst Case Time Complexity: O(log n)

Average Case Time Complexity: O(log n)

Space Complexity: O(1)

* **Discuss which algorithm is more suitable for your platform and why**

**Ans :**

Binary search will be more appropriate for an e-commerce platform with a large number of products if the product list is sorted, as it has lower time complexity. If the product list is supposed to be frequently updated, maintaining the sorted order is essential for binary search efficiency.