**Asymptotic Notation**

Big O Notation is used to analyze the time and space complexity of algorithms. It describes how the performance of an algorithm scales with the input size. For search operations, it helps in choosing the most efficient algorithm depending on the data size.

**Best, Average, and Worst-Case for Search Operations:**

* **Linear Search**:
  + Best Case: O(1) – Element is at the first position
  + Average Case: O(n/2) ≈ O(n)
  + Worst Case: O(n) – Element is at the end or not found
* **Binary Search** (requires sorted array):
  + Best Case: O(1)
  + Average Case: O(log n)
  + Worst Case: O(log n)

**Analysis**

**Time Complexity Comparison:**

* Linear Search: O(n)
* Binary Search: O(log n) (It requires sorted data)

**Algorithm Suitability:**

* Linear Search is simple and does not require sorting. Suitable for small or unsorted datasets.
* Binary Search is efficient and suitable for large datasets where data is sorted. It provides much faster search performance compared to linear search.

For an e-commerce platform handling a large number of products **, binary search** is a better choice due to its logarithmic time complexity.