**Understand Asymptotic Notation**

**Big O Notation and Its Importance in Analyzing Algorithms:**

Big O notation is a way to describe the performance or complexity of an algorithm as the input size grows. It helps us understand the worst-case time or space required by the algorithm. For search operations, Big O gives a clear idea of how fast or slow a search technique is, which helps us choose the right one for better performance in an e-commerce platform.

**Best, Average, and Worst-Case Scenarios:**

- **Best Case:** The condition where the desired product is found at the very beginning. For linear search, this is O(1).  
- **Average Case**: The product is somewhere in the middle. For linear search, it’s O(n/2) ≈ O(n).  
- **Worst Case:** The product is not present or at the end of the list. For linear search, this is O(n). For binary search, the worst case is O(log n), which is still much faster.

**Analysis**

**Time Complexity Comparison:**

- **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)

**Which Algorithm is More Suitable and Why:**

Binary search is more suitable for large-scale platforms because it is much faster than linear search. However, it requires the data to be sorted. For dynamic data where sorting isn’t possible each time, linear search might be used. But for optimized performance in most e-commerce applications where data is sorted or can be pre-sorted, binary search is preferred.