**Analysis – Part 1: Comparison of Time Complexity**

We compare the **time complexity** of **Linear Search** and **Binary Search** algorithms used in the search functionality of the e-commerce platform.

**Linear Search:**

**Description:**  
Linear search checks each product one by one from the beginning of the list until the desired product is found or the end of the list is reached.

**Time Complexity:**

* **Best Case:** O(1) – when the product is found at the very beginning of the list.
* **Average Case:** O(n) – when the product is located somewhere in the middle of the list.
* **Worst Case:** O(n) – when the product is at the end or not present in the list at all.

**Binary Search:**

**Description:**  
Binary search works only when the list of products is sorted. It repeatedly divides the list into halves and checks the middle element in each step to find the desired product.

**Time Complexity:**

* **Best Case:** O(1) – when the product is exactly at the middle position on the first check.
* **Average Case:** O(log n) – the list is divided in half each time, reducing the number of comparisons.
* **Worst Case:** O(log n) – even in the worst case, the number of steps increases logarithmically with the list size.

**Analysis – Part 2: Suitable Search Algorithm for the E-commerce Platform**

In an e-commerce platform, the search functionality plays a critical role in delivering fast and accurate results to users. Therefore, choosing the right search algorithm depends on the nature and structure of the product data.

**Suitability of Linear Search:**

Linear search is simple to implement and does not require the product list to be sorted. It can be useful for:

* Small datasets where the number of products is limited.
* Situations where the list changes frequently and keeping it sorted is difficult.
* Quick testing or debugging in development stages.

However, as the number of products grows, linear search becomes slower because it may need to check each item individually. This results in poor performance for large inventories.

**Suitability of Binary Search:**

Binary search is highly efficient and provides significantly faster results, especially for large datasets. It is suitable when:

* The product list is **sorted by product ID** or another searchable attribute.
* Fast performance and responsiveness are critical.
* The dataset is relatively stable or can be kept sorted during updates.

Since binary search operates in O(log n) time, it scales well with the size of the inventory and is ideal for real-time applications where users expect quick search results.