**Exercise 2: E-commerce Platform Search Function – Theoretical**

**1. Understanding Asymptotic Notation**

* Big O Notation is a mathematical concept used to describe how the performance of an algorithm changes as the size of input increases. It helps us understand how quickly the runtime or space requirements grow, which is very important for optimizing algorithms in real-world applications like an e-commerce platform.
* Big O focuses on the *worst-case* scenario to make sure that even under heavy load, the system will perform efficiently.
* For example, if a search algorithm has a time complexity of O(n), it means the time it takes to search increases linearly with the number of products. If it’s O(log n), the time grows much slower even as the product count increases.

**Best, Average, and Worst Cases in Search Operations:**

* **Best Case**: This is when the element is found in the first few steps. For linear search, the best case is when the item is at the beginning (O(1)).
* **Average Case**: This assumes that the element is somewhere in the middle. Linear search here takes around O(n/2), which simplifies to O(n).
* **Worst Case**: The element is either at the end or not present at all. For linear search, it’s O(n), and for binary search, it’s O(log n), because it keeps dividing the array in half.

**4. Analysis and Comparison**

* **Linear Search**:

Time Complexity: O(n)

Easy to implement and works even if data is not sorted.

Slower for large datasets, as it has to check each item.

* **Binary Search**:

Time Complexity: O(log n)

Much faster than linear search for large datasets.

Requires the array to be sorted before performing the search.