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

**🔹 1. Big O Notation**

Big O notation is a way to represent the efficiency of an algorithm in terms of how its running time or space grows as input size increases. It helps us compare and select the best algorithm based on performance.

In search operations:

* **Best Case**: When the item is found immediately.
* **Average Case**: Based on average position or distribution.
* **Worst Case**: When the item is last or not found at all.

Examples:

* **Linear Search**: Best – O(1), Worst – O(n)
* **Binary Search**: Best – O(1), Worst – O(log n)

**🔹 2. Product Class Setup**

We create a Product class with the following attributes:

* productId: Unique identifier
* productName: Used in search queries
* category: Represents the product type (e.g., Electronics, Fashion)

**🔹 3. Implementation of Search**

Two search algorithms are implemented:

* **Linear Search**: Goes through each product one by one. Simple, but slow for large lists. Works even on unsorted data.
* **Binary Search**: Divides the list and eliminates half the search space at each step. Requires the array to be sorted by productName. Much faster for large data.

**🔹 4. Analysis and Recommendation**

* **Linear Search**: Easy to implement, no sorting needed. Suitable for small datasets or quick tests.
* **Binary Search**: Efficient for large and sorted datasets. Recommended for scalable platforms like e-commerce systems.

In a real e-commerce application, where thousands of products are listed and fast response is important, **binary search** or even more advanced methods (like hash maps or search indexing) are preferred.