**📊 E-commerce Search Function – Analysis & Explanation**

This document provides theoretical insights as required in the assignment, covering asymptotic notation, search algorithm performance, and analysis of which search method is better suited for an e-commerce platform.

**1. 🧠 Understanding Asymptotic Notation**

**What is Big O Notation?**

Big O notation is used to describe the **time complexity** or **performance** of an algorithm as the input size grows. It helps us estimate how fast or slow an algorithm will be, and compare multiple approaches.

**Why is it useful?**

* Predicts performance before execution.
* Helps choose efficient algorithms.
* Makes software scalable.

**2. 🔍 Best, Average, and Worst-Case Scenarios for Search**

| **Case** | **Description** |
| --- | --- |
| **Best Case** | The desired product is found immediately. |
| **Average Case** | The product is somewhere in the middle. |
| **Worst Case** | The product is not found or is the last in the array. |

**Time Complexities:**

**Linear Search:**

* **Best Case:** O(1)
* **Average Case:** O(n)
* **Worst Case:** O(n)

**Binary Search (on sorted array):**

* **Best Case:** O(1)
* **Average Case:** O(log n)
* **Worst Case:** O(log n)

**3. 🛠️ Setup (Code Summary)**

We created a Product class with:

* productId (int)
* productName (String)
* category (String)

Then, we implemented:

* Linear Search (unsorted array)
* Binary Search (sorted array)

**4. ⚖️ Time Complexity Comparison**

| **Algorithm** | **Time Complexity** | **Sorted Data Required?** |
| --- | --- | --- |
| Linear Search | O(n) | ❌ No |
| Binary Search | O(log n) | ✅ Yes |

**5. ✅ Which Algorithm is More Suitable?**

For an **e-commerce platform** with large product data:

* **Binary Search** is better in performance due to O(log n) time.
* However, the array must be **sorted**, which adds a small preprocessing cost.

So, if data is sorted by productName or productId, **Binary Search** is highly recommended for **faster search**.

📌 **Conclusion**:

Binary search is more efficient and scalable. Linear search can be used as a fallback for unsorted or small data collections.