**Project Title: ECommerceSearchOptimization**

**Objective:**

Implement and compare Linear Search and Binary Search to determine which algorithm is more efficient for product searching on an e-commerce platform.

**Key Concepts:**

* Big O Notation is used to analyze algorithm efficiency
* Search operations are analyzed in best, average, and worst-case time complexity
* Binary Search is faster but requires a sorted dataset

**Understand Asymptotic Notation :**

What is Big O Notation?

Big O Notation describes the upper bound of an algorithm’s running time as the input size n grows:

* Helps evaluate scalability and performance
* Focuses on dominant terms, ignoring constants

Best, Average, Worst Case in Search –

| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** | |
| --- | --- | --- | --- | --- |
| Linear Search | O(1) | O(n/2 | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |

**Product Class Attributes:**

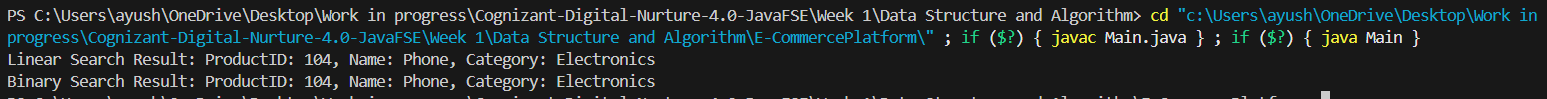
* productId: Unique ID
* productName: Item name
* category: Product category

**Implemented Algorithms:**

* **Linear Search**: Simple sequential scan — O(n)
* **Binary Search**: Efficient divide-and-conquer — O(log n)

**Code Components:**

* Product.java: Data model
* SearchAlgorithms.java: Logic for searching and sorting
* Main.java: Test class

**Output Example:**

**Conclusion:**

For optimized performance on large and frequently searched datasets, Binary Search is preferred — provided data is sorted. Otherwise, for simple and small searches, Linear Search works well.