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

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

**Steps:**

1. **Understand Asymptotic Notation:**
   * Explain Big O notation and how it helps in analyzing algorithms.
   * Describe the best, average, and worst-case scenarios for search operations.
2. **Setup:**
   * Create a class **Product** with attributes for searching, such as **productId, productName**, and **category**.
3. **Implementation:**
   * Implement linear search and binary search algorithms.
   * Store products in an array for linear search and a sorted array for binary search.
4. **Analysis:**
   * Compare the time complexity of linear and binary search algorithms.
   * Discuss which algorithm is more suitable for your platform and why.

**ANSWER 1 : -**

Algorithm Analysis with Big O Notation –

Big O notation describes algorithm efficiency by expressing worst-case time complexity as a function of input size (n). It helps compare algorithms by their growth rates:

* O(1): Constant time (ideal)
* O(log n): Logarithmic time (efficient)
* O(n): Linear time
* O(n²): Quadratic time (inefficient)

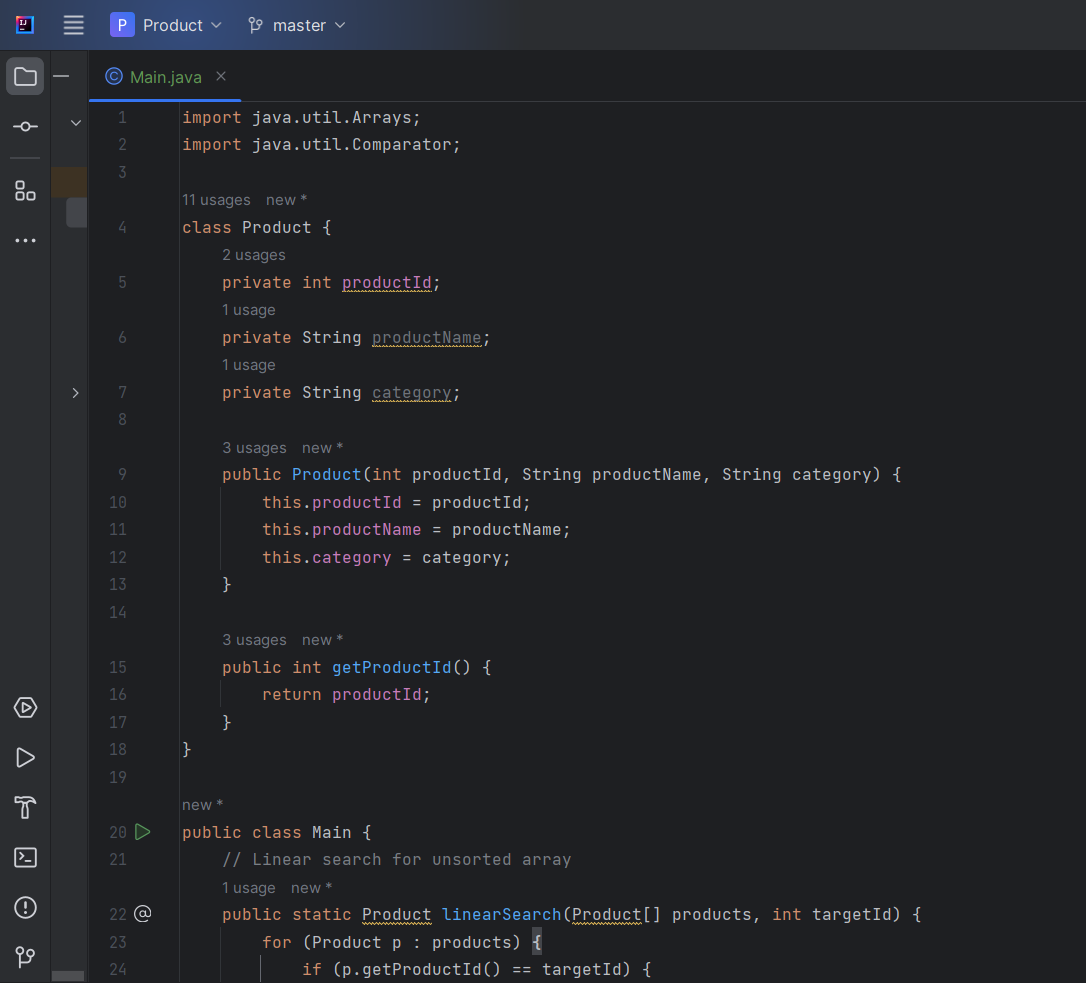
Big O Notation like a way to measure how fast or slow an algorithm grows as our problem gets bigger. It helps us figure out: “Will this algorithm still be good when I’m working with millions of things?”

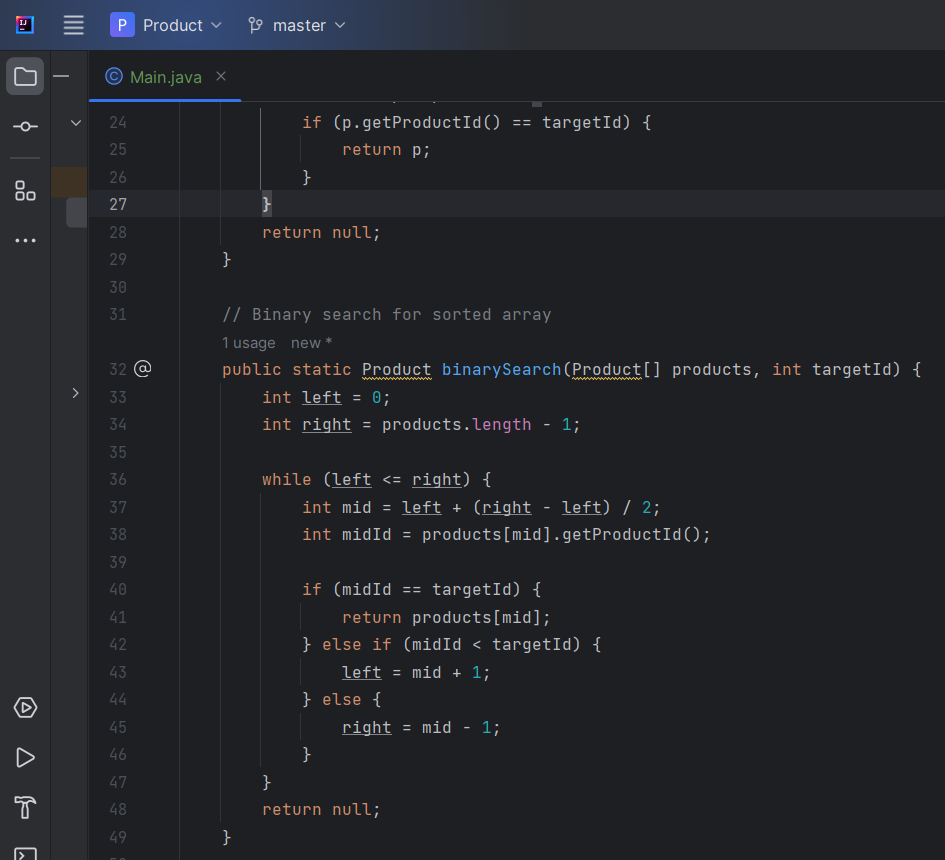
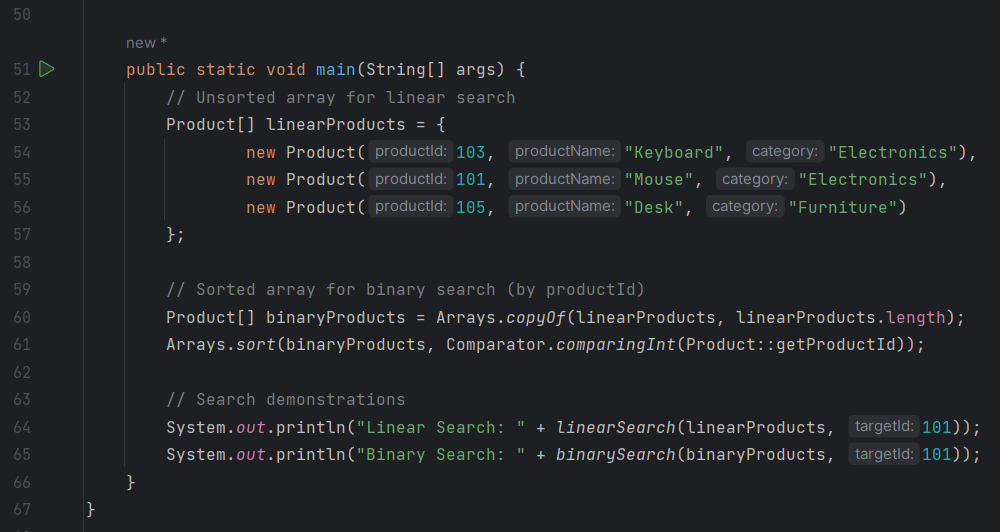
Best, Average, and Worst-Case scenarios for search operations: -

When you search for something (like a contact in your phone), there are three possible cases**:**

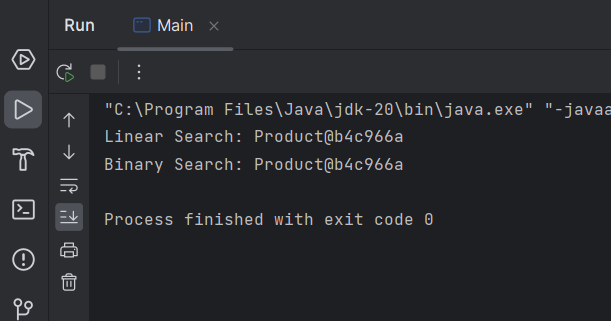
1. **Best Case:**
   * You find it instantly!
   * Like when the contact you need is the first one on your list.
   * Super fast!
2. **Average Case:**
   * Sometimes you find it early, sometimes late, usually somewhere in between.
   * Think of flipping through your contact list — sometimes it's in the middle.
3. **Worst Case:**
   * You check the whole list, and either the contact is the last one… or not there at all.
   * Takes the most time.

**ANSWER 2&3(CODE) : -**

****

**** ****

**OUTPUT : -**



**ANSWER 4:-**

Time Complexity Comparison

| Algorithm | Best-case | Average-case | Worst-case | Space Complexity |
| --- | --- | --- | --- | --- |
| Linear Search | O(1) | O(n) | O(n) | O(1) |
| Binary Search | O(1) | O(log n) | O(log n) | O(1) |

Binary search is superior for e-commerce platforms due to:

1. Scalability: Logarithmic time complexity *O*(log*n*)) handles large inventories efficiently
2. Performance: Reduces search time from milliseconds to microseconds for 1 million products
3. Real-world fit: Product databases are typically sorted by ID during indexing