**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.

**Source Code :**

**import java.util.Arrays;**

**import java.util.Comparator;**

**public class ECommerceSearch {**

**public static void main(String[] *args*) {**

**Product[] products = {**

**new Product(101, "Laptop", "Electronics"),**

**new Product(102, "Shirt", "Clothing"),**

**new Product(103, "Mobile", "Electronics"),**

**new Product(104, "Shoes", "Footwear"),**

**new Product(105, "Book", "Stationery")**

**};**

**Product target = products[2];**

**int linearIndex = Search.*linearSearch*(products, target.getProductId());**

**System.out.println("Linear Search Index: " + linearIndex);**

**System.out.println("Product found : "+ products[linearIndex].toString());**

**Arrays.*sort*(products, Comparator.*comparingInt*(Product::getProductId));**

**int binaryIndex = Search.*binarySearch*(products, target.getProductId());**

**System.out.println("Binary Search Index: " + binaryIndex);**

**System.out.println("Product found : "+ products[binaryIndex].toString());**

**}**

**}**

**class Product {**

**private int productId;**

**private String productName;**

**private String category;**

**public Product(int *productId*, String *productName*, String *category*) {**

**this.productId = *productId*;**

**this.productName = *productName*;**

**this.category = *category*;**

**}**

**public int getProductId() {**

**return productId;**

**}**

**public String toString() {**

**return productId + " - " + productName + " - " + category;**

**}**

**}**

**class Search {**

**public static int linearSearch(Product[] *products*, int *productId*) {**

**for (int i = 0; i < *products*.length; i++) {**

**if (*products*[i].getProductId() == *productId*) return i;**

**}**

**return -1;**

**}**

**public static int binarySearch(Product[] *products*, int *productId*) {**

**int left = 0, right = *products*.length - 1;**

**while (left <= right) {**

**int mid = left + (right - left) / 2;**

**if (*products*[mid].getProductId() == *productId*) return mid;**

**if (*products*[mid].getProductId() < *productId*) left = mid + 1;**

**else right = mid - 1;**

**}**

**return -1;**

**}**

**}**

Output :

