**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 :**

**package week1.algorithmsanddatastructures;**

**import java.util.Arrays;**

**import java.util.Comparator;**

**public class ecommerce\_search {**

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

**Product[] itemList = {**

**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 searchItem = itemList[2];**

**int indexLinear = Search.linearSearch(itemList, searchItem.getProductId());**

**System.out.println("Result from Linear Search -> Index: " + indexLinear);**

**System.out.println("Product located: " + itemList[indexLinear].toString());**

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

**int indexBinary = Search.binarySearch(itemList, searchItem.getProductId());**

**System.out.println("Result from Binary Search -> Index: " + indexBinary);**

**System.out.println("Product located: " + itemList[indexBinary].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[] list, int targetId) {**

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

**if (list[i].getProductId() == targetId) return i;**

**}**

**return -1;**

**}**

**public static int binarySearch(Product[] list, int targetId) {**

**int low = 0, high = list.length - 1;**

**while (low <= high) {**

**int mid = low + (high - low) / 2;**

**if (list[mid].getProductId() == targetId) return mid;**

**if (list[mid].getProductId() < targetId) low = mid + 1;**

**else high = mid - 1;**

**}**

**return -1;**

**}**

**}**

Output :

