**Algorithms and data structures:**

**Ex 2. E-commerce platform search function:**

**Code:**

import java.util.\*;

public class ProductSearchDemo {

// Step 2: Product class

static class Product implements Comparable<Product> {

private final int productId;

private final String productName;

private final String category;

public Product(int id, String name, String category) {

this.productId = id;

this.productName = name;

this.category = category;

}

public int getProductId() { return productId; }

@Override

public int compareTo(Product o) {

return Integer.compare(this.productId, o.productId);

}

@Override

public String toString() {

return "[" + productId + "] " + productName + " (" + category + ")";

}

}

// Step 3: Linear Search (O(n))

static Product linearSearch(Product[] products, int targetId, int[] comparisons) {

for (Product p : products) {

comparisons[0]++;

if (p.getProductId() == targetId) return p;

}

return null;

}

// Step 3: Binary Search (O(log n)) — on sorted array

static Product binarySearch(Product[] sorted, int targetId, int[] comparisons) {

int low = 0, high = sorted.length - 1;

while (low <= high) {

int mid = (low + high) >>> 1;

comparisons[0]++;

int midId = sorted[mid].getProductId();

if (midId == targetId) return sorted[mid];

else if (midId < targetId) low = mid + 1;

else high = mid - 1;

}

return null;

}

// Step 4: Demo

public static void main(String[] args) {

Product[] catalog = {

new Product(104, "Keyboard", "Electronics"),

new Product(101, "Sneakers", "Footwear"),

new Product(109, "Coffee Beans", "Grocery"),

new Product(105, "Desk Lamp", "Home Decor"),

new Product(117, "Gaming Mouse", "Electronics"),

new Product(110, "Water Bottle", "Sports")

};

int targetId = 109;

int[] cmpLinear = {0};

long t1 = System.nanoTime();

Product resultLinear = linearSearch(catalog, targetId, cmpLinear);

long t2 = System.nanoTime();

Product[] sortedCatalog = catalog.clone();

Arrays.sort(sortedCatalog);

int[] cmpBinary = {0};

long t3 = System.nanoTime();

Product resultBinary = binarySearch(sortedCatalog, targetId, cmpBinary);

long t4 = System.nanoTime();

System.out.println("Linear Search Result: " + resultLinear);

System.out.println("Linear Search Comparisons: " + cmpLinear[0]);

System.out.println("Linear Search Time (ns): " + (t2 - t1));

System.out.println("Binary Search Result: " + resultBinary);

System.out.println("Binary Search Comparisons: " + cmpBinary[0]);

System.out.println("Binary Search Time (ns): " + (t4 - t3));

}

}

Output:

