# Module 2 – Data Structures and Algorithms

# Excercise 2 - E-commerce Platform Search Function

import java.util.Arrays;

import java.util.Comparator;

public class Main {

public static void main(String[] args) {

Product[] products = {

new Product(205, "Rice Bag", "Grocery"),

new Product(201, "Toothpaste", "Personal Care"),

new Product(203, "Detergent", "Cleaning"),

new Product(202, "Notebook", "Stationery"),

new Product(204, "Vegetable Oil", "Grocery")

};

System.out.println("Linear Search:");

Product result1 = Search.linearSearch(products, 209);

System.out.println(result1 != null ? result1 : "Product not found");

System.out.println("Binary Search:");

Search.sortById(products);

Product result2 = Search.binarySearch(products, 202);

System.out.println(result2 != null ? result2 : "Product not found");

}

public static class Product {

int productId;

String productName;

String category;

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

this.productId = id;

this.productName = name;

this.category = category;

}

public String toString() {

return "ID: " + productId + " | Name: " + productName + " | Category: " + category;

}

}

public static class Search {

static Product linearSearch(Product[] items, int targetId) {

for (Product p : items) {

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

}

return null;

}

static Product binarySearch(Product[] items, int targetId) {

int left = 0, right = items.length - 1;

while (left <= right) {

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

int id = items[mid].productId;

if (id == targetId) return items[mid];

if (id < targetId) left = mid + 1;

else right = mid - 1;

}

return null;

}

static void sortById(Product[] items) {

Arrays.sort(items, Comparator.comparingInt(p -> p.productId));

}

}

}

