Week 2 Data structure

Exercise2: **E-commerce Platform Search Function**

Product.java

public class Product {

int productId;

String productName;

String category;

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

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString() {

return "Product [ID=" + productId + ", Name=" + productName + ", Category=" + category + "]";

}

}

SearchDemo.java

import java.util.Arrays;

import java.util.Comparator;

public class SearchDemo {

public static Product linearSearch(Product[] products, String targetName) {

for (Product product : products) {

if (product.productName.equalsIgnoreCase(targetName)) {

return product;

}

}

return null;

}

public static Product binarySearch(Product[] products, String targetName) {

Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

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

while (left <= right) {

int mid = (left + right) / 2;

int cmp = targetName.compareToIgnoreCase(products[mid].productName);

if (cmp == 0) return products[mid];

else if (cmp < 0) right = mid - 1;

else left = mid + 1;

}

return null;

}

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop", "Electronics"),

new Product(2, "Shampoo", "Personal Care"),

new Product(3, "T-shirt", "Clothing"),

new Product(4, "Phone", "Electronics"),

new Product(5, "Book", "Education"),

new Product(6, "Headphones", "Electronics")

};

// Linear Search

Product result1 = linearSearch(products, "Phone");

System.out.println("Linear Search Result: " + result1);

// Binary Search

Product result2 = binarySearch(products, "Phone");

System.out.println("Binary Search Result: " + result2);

}

}

