**Exercise 2: E-commerce Platform Search Function**

Big O notation helps us understand how efficient an algorithm is, especially when the number of items grows. It describes how the time taken by an algorithm increases as the input size increases, without getting into specific hardware or exact timing.

For example:

* **O(1)**: Constant time.
* **O(n)**: Linear time.
* **O(log n)**: Logarithmic time.

**Product.java**

class Product {

int productId;

String productName;

String category;

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

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString() {

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

}

}

**ECommerceSearch.java**

import java.util.Arrays;

import java.util.Comparator;

public class ECommerceSearch {

// Linear Search: O(n)

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

for (Product product : products) {

if (product.productId == targetId) {

return product;

}

}

return null;

}

// Binary Search: O(log n)

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

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

while (left <= right) {

int mid = (left + right) / 2;

if (products[mid].productId == targetId) {

return products[mid];

} else if (products[mid].productId < targetId) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

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

new Product(102, "Watch", "Accessories"),

new Product(101, "T-shirt", "Clothing"),

new Product(105, "Smartphone", "Electronics"),

new Product(103, "Shoes", "Footwear")

};

// Linear Search Test

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

Product foundLinear = linearSearch(products, 105);

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

// Sort products for binary search by productId

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

// Binary Search Test

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

Product foundBinary = binarySearch(products, 105);

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

}

}

**OUTPUT SCREENSHOT**:

