**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;  
  
 public Product(int productId, String productName, String category) {  
 this.productId = productId;  
 this.productName = productName;  
 this.category = category;  
 }  
}

**SearchUtility.java**

class SearchUtility {  
  
 public static Product linearSearch(Product[] products, String targetName) {  
 for (Product product : products) {  
 if (product.productName.equalsIgnoreCase(targetName)) {  
 return product;  
 }  
 }  
 return null;  
 }  
  
 // Simple bubble sort based on productName (case-insensitive)  
 public static void sortProductsByName(Product[] products) {  
 int n = products.length;  
 for (int i = 0; i < n - 1; i++) {  
 for (int j = 0; j < n - i - 1; j++) {  
 if (products[j].productName.compareToIgnoreCase(products[j + 1].productName) > 0) {  
 Product temp = products[j];  
 products[j] = products[j + 1];  
 products[j + 1] = temp;  
 }  
 }  
 }  
 }  
  
 public static Product binarySearch(Product[] products, String targetName) {  
 *sortProductsByName*(products);  
 int low = 0, high = products.length - 1;  
 while (low <= high) {  
 int mid = (low + high) / 2;  
 int cmp = products[mid].productName.compareToIgnoreCase(targetName);  
 if (cmp == 0) return products[mid];  
 else if (cmp < 0) low = mid + 1;  
 else high = mid - 1;  
 }  
 return null;  
 }  
  
  
}

**ECommercePlatformSearch.java**

public class EcommercePlatformSearch {  
  
 public static void main(String[] args) {  
 Product[] products = {  
 new Product(101, "Laptop", "Electronics"),  
 new Product(102, "Shoes", "Footwear"),  
 new Product(103, "Mobile", "Electronics"),  
 new Product(104, "Watch", "Accessories"),  
 new Product(105, "Headphones", "Electronics")  
 };  
  
 String searchName = "Watch";  
  
 Product result1 = SearchUtility.*linearSearch*(products, searchName);  
  
 if(result1!=null)  
 {  
  
 System.*out*.println("The result returned by linear search ");  
 result1.display();  
 }  
 else  
 {  
 System.*out*.println("Product Not Found");  
 }  
  
  
  
  
 Product result2 = SearchUtility.*binarySearch*(products, searchName);  
  
 if(result2!=null)  
 {  
 System.*out*.println("The result returned by binary search ");  
 result2.display();  
 }  
 else  
 {  
 System.*out*.println("Product Not Found");  
 }  
  
 }  
}

**OUTPUT SCREENSHOT**:

