Exercise 2: E-commerce Platform Search Function

# Scenario:

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

# Steps:

## Step 1: Understand Asymptotic Notation

Big O notation is used to describe the performance or complexity of an algorithm.   
It provides an upper bound on the running time in the worst-case scenario as the input size grows.  
- Best Case: The scenario where the algorithm performs the minimum number of steps (e.g., item is found at the beginning).  
- Average Case: The expected number of steps for a random input.  
- Worst Case: The scenario where the algorithm performs the maximum number of steps (e.g., item not found or at the end).

## Step 2: Setup

// File: Product.java  
package search;  
  
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;  
 }  
  
 public String toString() {  
 return productId + ": " + productName + " (" + category + ")";  
 }  
}

## Step 3: Implementation

// File: SearchAlgorithms.java  
package search;  
  
public class SearchAlgorithms {  
  
 // Linear Search  
 public static Product linearSearch(Product[] products, String targetName) {  
 for (Product product : products) {  
 if (product.productName.equalsIgnoreCase(targetName)) {  
 return product;  
 }  
 }  
 return null;  
 }  
  
 // Binary Search (Assumes sorted array by productName)  
 public static Product binarySearch(Product[] products, String targetName) {  
 int left = 0, right = products.length - 1;  
 while (left <= right) {  
 int mid = (left + right) / 2;  
 int cmp = products[mid].productName.compareToIgnoreCase(targetName);  
 if (cmp == 0) return products[mid];  
 else if (cmp < 0) left = mid + 1;  
 else right = mid - 1;  
 }  
 return null;  
 }  
}

## Step 4: Analysis

- Linear Search: Time Complexity is O(n). It scans each element until the target is found.  
- Binary Search: Time Complexity is O(log n). It repeatedly divides the search interval in half.  
  
Binary search is faster but requires the array to be sorted. For a large, sorted dataset, binary search is preferred due to its logarithmic complexity.   
Linear search is suitable for small or unsorted datasets.