

Exercise 2: E-commerce Platform Search Function

Understanding Asymptotic Notation:

What is Big O Notation?

- Big O notation is used to describe how the performance of an algorithm scales with the size of the input.
- It gives an upper bound on the number of steps an algorithm takes in the worst case.

Big O for Search Algorithms

Search Type	Best Case	Average Case	Worst Case
Linear Search	$O(1)$	$O(n)$	$O(n)$
Binary Search	$O(1)$	$O(\log n)$	$O(\log n)$

Product Class

```
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 productId + " - " + productName + " [" + category + "];"  
    }  
}
```

SEARCH CLASS:

```
public class Search {  
  
    public static Product linearSearch(Product[] products, int targetId) {  
        for (Product product : products) {  
            if (product.productId == targetId) {  
                return product;  
            }  
        }  
        return null;  
    }  
  
    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;  
    }  
}
```

MAIN CLASS:

```
import java.util.Arrays;  
import java.util.Comparator;
```

```
public class Main {  
    public static void main(String[] args) {  
        Product[] products = {  
            new Product(105, "Phone", "Electronics"),  
            new Product(102, "Shirt", "Clothing"),  
            new Product(108, "Book", "Education"),  
            new Product(101, "Laptop", "Electronics"),  
            new Product(104, "Shoes", "Footwear")  
        };  
  
        // Sort for binary search  
        Arrays.sort(products, Comparator.comparingInt(p -> p.productId));  
  
        int searchId = 104;  
  
        // Linear Search  
        Product result1 = Search.linearSearch(products, searchId);  
        System.out.println("Linear Search: " + (result1 != null ? result1 : "Product not found"));  
  
        // Binary Search  
        Product result2 = Search.binarySearch(products, searchId);  
        System.out.println("Binary Search: " + (result2 != null ? result2 : "Product not found"));  
    }  
}
```