**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:**

1. **Understand Asymptotic Notation:**
   * Explain Big O notation and how it helps in analyzing algorithms.
   * Describe the best, average, and worst-case scenarios for search operations.
2. **Setup:**
   * Create a class CatalogItem with attributes like catalogId, name, and type.
3. **Implementation:**
   * Implement **linear search** and **binary search** algorithms.
   * Use an **unsorted array** for linear search.
   * Use a **sorted array** for binary search (sorted by catalogId).
4. **Analysis:**
   * Compare the time complexity of linear and binary search algorithms.
     + - * Linear Search: **O(n)**
         * Binary Search: **O(log n)**
   * Binary search is faster for large sorted datasets, while linear search is better for small or unsorted lists.

**Source Code :**

package week1.algorithmsanddatastructures;  
  
import java.util.Arrays;  
import java.util.Comparator;  
  
public class MarketplaceLookup  
{  
 public static void main(String[] args)  
 {  
 CatalogItem[] catalog = {  
 new CatalogItem(301, "Camera", "Photography"),  
 new CatalogItem(302, "T-Shirt", "Apparel"),  
 new CatalogItem(303, "Smartwatch", "Electronics"),  
 new CatalogItem(304, "Boots", "Footwear"),  
 new CatalogItem(305, "Planner", "Office Supplies")  
 };  
  
 CatalogItem searchItem = catalog[2];  
 int linearResult = Locator.linearSearch(catalog, searchItem.getCatalogId());  
 System.out.println("Linear Search Result Index: " + linearResult);  
 System.out.println("Item found: " + catalog[linearResult].toString());  
  
 Arrays.sort(catalog, Comparator.comparingInt(CatalogItem::getCatalogId));  
 int binaryResult = Locator.binarySearch(catalog, searchItem.getCatalogId());  
 System.out.println("Binary Search Result Index: " + binaryResult);  
 System.out.println("Item found: " + catalog[binaryResult].toString());  
 }  
}  
  
class CatalogItem  
{  
 private int catalogId;  
 private String name;  
 private String type;  
  
 public CatalogItem(int catalogId, String name, String type)  
 {  
 this.catalogId = catalogId;  
 this.name = name;  
 this.type = type;  
 }  
  
 public int getCatalogId()  
 {  
 return catalogId;  
 }  
  
 public String toString()  
 {  
 return catalogId + " - " + name + " - " + type;  
 }  
}  
  
class Locator  
{  
 public static int linearSearch(CatalogItem[] catalog, int catalogId)  
 {  
 for (int i = 0; i < catalog.length; i++)  
 {  
 if (catalog[i].getCatalogId() == catalogId) return i;  
 }  
 return -1;  
 }  
  
 public static int binarySearch(CatalogItem[] catalog, int catalogId)  
 {  
 int low = 0, high = catalog.length - 1;  
 while (low <= high)  
 {  
 int mid = low + (high - low) / 2;  
 if (catalog[mid].getCatalogId() == catalogId) return mid;  
 if (catalog[mid].getCatalogId() < catalogId) low = mid + 1;  
 else high = mid - 1;  
 }  
 return -1;  
 }  
}

Output :

Linear Search Index: 2

Product found : 103 - Mobile - Electronics

Binary Search Index: 2

Product found : 103 - Mobile - Electronics