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

**Asymptotic Notation (Big O)**

* Big O describes worst-case runtime/space complexity.
* **Best-case**: Minimum effort (e.g., item is first).
* **Average-case**: Expected runtime over all possible inputs.
* **Worst-case**: Maximum effort (e.g., item is last or not found).

**Search Complexities:**

| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| --- | --- | --- | --- |
| Linear Search | O(1) | O(n) | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |

Solution 🡪

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;

}

}

import java.util.Arrays;

import java.util.Comparator;

public class ProductSearch {

// Linear Search

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

for (Product p : products) {

if (p.productId == targetId) return p;

}

return null;

}

// Binary Search (on sorted array)

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];

if (products[mid].productId < targetId) left = mid + 1;

else right = mid - 1;

}

return null;

}

public static void sortByProductId(Product[] products) {

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

}

}

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(101, "Laptop", "Computers"),

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

new Product(102, "Smartphone", "Mobiles")

};

System.out.println("🔍 Linear Search:");

Product found = ProductSearch.linearSearch(products, 103);

System.out.println(found != null ? found : "Not Found");

System.out.println("\n🔍 Binary Search:");

ProductSearch.sortByProductId(products); // Required for binary search

Product found2 = ProductSearch.binarySearch(products, 103);

System.out.println(found2 != null ? found2 : "Not Found");

}

}

**Performance Conclusion**

* Use Linear Search for small datasets or unsorted data.
* Use Binary Search when:
  + Data is sorted by key (e.g., productId)
  + You need fast, consistent lookup (logarithmic time)