**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 optimised for fast performance.

**Steps:**

***Understand Asymptotic Notation:***

1. Explain Big O notation and how it helps in analysing algorithms.

***Ans:*** Big O notation is a mathematical notation used to describe the upper bound of an algorithm's running time. It gives us an idea of the worst-case scenario in terms of the number of operations an algorithm will perform as the input size grows. It helps in analysing the efficiency and scalability of algorithms by abstracting away constants and less significant terms.

1. Describe the best, average, and worst-case scenarios for search operations.

**Best Case:** The scenario where the algorithm performs the minimum number of operations. For example, in a search operation, this would be finding the element in the first position.

**Average Case:** The scenario representing the expected number of operations for typical input. This is often a weighted average of all possible cases.

**Worst Case:** The scenario where the algorithm performs the maximum number of operations. For a search operation, this would be searching through all elements without finding the target.

***Setup:***

Create a class Product with attributes for searching, such as productId, productName, and category.

Code: Product.java

***Implementation:***

Implement linear search and binary search algorithms.

Store products in an array for linear search and a sorted array for binary search.

Code: Product,java

***Analysis:***

1. Compare the time complexity of linear and binary search algorithms.

Ans; **Linear Search:** O(n) in the worst case, where n is the number of products.

**Binary Search:** O(log n) in the worst case, where n is the number of products.

1. Discuss which algorithm is more suitable for your platform and why.

Ans: **Linear Search** is suitable for small arrays or unsorted data. It's simple to implement and does not require the data to be sorted.

**Binary Search** is more efficient for large arrays with sorted data. It significantly reduces the number of comparisons needed to find an element.