**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.

**1. Understanding Asymptotic Notation:**

**Big O Notation:**

Big O notation describes how the runtime or space requirement of an algorithm grows with the size of the input. It helps in analyzing and comparing algorithms to choose the most efficient one.

Examples:

* O(1) – Constant Time
* O(n) – Linear Time
* O(log n) – Logarithmic Time

**Best, Average, and Worst-Case Scenarios:**

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

* Linear Search: Checks every element one by one until the element is found or the end of the array is reached.
* Binary Search: Divides the array into halves to locate the element faster but requires sorted data.

**2. Setup (in code folder)**

**3. Implementation (in code folder)**

**4. Analysis:**

| **Algorithm** | **Time Complexity** | **Space Complexity** |
| --- | --- | --- |
| Linear Search | O(n) | O(1) |
| Binary Search | O(log n) | O(1) |

* Linear Search checks each product one by one, leading to O(n) time complexity in the worst case.
* Binary Search divides the search range, achieving O(log n) time complexity but requires the array to be sorted.

**Comparison between Linear Search and Binary Search:**

* **Linear Search**: Simple to implement and works with unsorted data. Suitable for small datasets.
* **Binary Search**: Faster for large datasets but needs a sorted array before searching.
* As the size of the dataset increases, Binary Search performs better in terms of speed compared to Linear Search.