# Exercise 2: E-commerce Platform Search Function

## 1. Understand Asymptotic Notation

Big O Notation:  
Big O notation describes the upper bound of an algorithm's running time in relation to the input size. It provides a way to evaluate algorithm efficiency, especially when inputs become large.

Search Operation Scenarios:

- Best Case: The item is found in the first comparison.  
- Average Case: The item is somewhere in the middle.  
- Worst Case: The item is not found or is at the end of the search range.

Examples:

- Linear Search:  
 - Best: O(1)  
 - Average: O(n)  
 - Worst: O(n)

- Binary Search (on sorted data):  
 - Best: O(1)  
 - Average: O(log n)  
 - Worst: O(log n)

## 2. Setup

Product Class Design:  
Design a class named 'Product' with attributes like productId, productName, and category. These attributes are essential for implementing the search functionality.

## 3. Implementation

Implement two search algorithms:  
- Linear Search: Scan each product one by one and compare the product name.  
- Binary Search: Requires the array to be sorted by product name and uses a divide-and-conquer method to search efficiently.

## 4. Analysis

Time Complexity Comparison:

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)

- Linear Search is simple and works on unsorted data but becomes inefficient as the dataset grows.  
- Binary Search is significantly faster for large datasets but requires sorted data, which adds an O(n log n) sorting cost initially.

## Conclusion

For small product lists or rarely updated data, linear search might suffice. However, in a real-world e-commerce platform with thousands or millions of products, binary search (or more advanced structures like hash maps or search indexes) is more suitable due to its logarithmic time complexity after sorting.