**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.

**Step 1:** Understand Asymptotic Notation

**Big O Notation**

* **Concept**: Big O notation is used to describe the upper bound of the time complexity of an algorithm, giving an idea of its worst-case scenario in terms of the input size n. It helps to compare the efficiency of different algorithms.
* **Purpose**: Helps in analyzing and comparing the performance of algorithms, particularly as the input size grows.

**Scenarios for Search Operations**

* **Best Case**: The scenario where the search element is found at the beginning of the dataset.
* **Average Case**: The scenario where the search element is found somewhere in the middle of the dataset.
* **Worst Case**: The scenario where the search element is found at the end of the dataset or not found at all.

**Step 2:** Setup

**Create a Class Product :** Define a Product class with attributes productId, productName, and category.

### Step 3: Implementation

#### Implement Linear Search:

Linear search involves iterating through the array to find the target element.

**Implement Binary Search:**

Binary search involves dividing the array into halves to find the target element. The array must be sorted.

**Step 4:** Analysis

**Compare the Time Complexity of Linear and Binary Search**

* **Linear Search**:
  + **Best Case**: O(1) - When the target element is the first element.
  + **Average Case**: O(n) - On average, it checks half of the elements.
  + **Worst Case**: O(n) - When the target element is the last element or not present.
* **Binary Search**:
  + **Best Case**: O(1)- When the target element is the middle element.
  + **Average Case**: O(logn) - On average, it divides the array and reduces the search space by half each time.
  + **Worst Case**: O(logn) - When the target element is not present, it will still reduce the search space by half each time.

**Discuss Which Algorithm is More Suitable for Your Platform and Why**

* **Linear Search**:
  + **Advantages**: Simple to implement and does not require the array to be sorted.
  + **Disadvantages**: Inefficient for large datasets due to its O(n) time complexity.
  + **Use Case**: Suitable for small datasets or unsorted arrays where sorting is not feasible.
* **Binary Search**:
  + **Advantages**: Much more efficient for large datasets due to its O(logn) time complexity.
  + **Disadvantages**: Requires the array to be sorted, and the overhead of sorting needs to be considered if the array is frequently modified.
  + **Use Case**: Suitable for large datasets with infrequent modifications, where search efficiency is crucial.

For an e-commerce platform, **binary search** is generally more suitable because it provides much faster search performance for large datasets. However, it requires the dataset to be sorted, so it need to ensure that the product list is maintained in sorted order or sort it before performing the search.