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.

Steps:

1. **Understand Asymptotic Notation:**

   Explain Big O notation and how it helps in analyzing algorithms.
   Big O Notation:
   Big O notation is a mathematical way to describe the performance of an algorithm as the input size grows. It focuses on the upper bound, meaning the worst-case growth rate of time or space required.
   - It helps you understand how efficiently your code will run, especially when handling large datasets.
   - Big O doesn't care about exact execution time but how the time or space scales with input.
   - A lower Big O value typically indicates a more efficient algorithm.
     For example:
     - O(1) is constant time – fastest.
     - O(n) grows linearly with input.
     - O(log n) is logarithmic – very efficient for large data.

   Describe the best, average, and worst-case scenarios for search operations.
   Linear Search:
   Searches through each element in the list until it finds the match.

- Best Case:
  The item you're looking for is the very first element.
  Time Complexity: O(1)
- Average Case:
  The item is somewhere in the middle.
  Time Complexity: O(n)
- Worst Case:
  The item is the last one or not present at all.
  Time Complexity: O(n)

Binary Search:
Only works on sorted data. It divides the search space in half at each step.

- Best Case:
  The item is exactly in the middle.
  Time Complexity: O(1)
- Average Case:
  The item is somewhere in the array, and the algorithm halves the search range

repeatedly.
Time Complexity: O(log n)

- Worst Case:
  The item is at the extreme ends or not present.
  Time Complexity: O(log n)

2. **Setup:**
   - Create a class **Product** with attributes for searching, such as **productId, productName**, and **category**.
3. **Implementation:**
   - Implement linear search and binary search algorithms.
   - Store products in an array for linear search and a sorted array for binary search.
4. **Analysis:**
   - Compare the time complexity of linear and binary search algorithms.

Scenario	Linear Search	Binary Search
Best Case	O(1)	O(1)
Average	O(n)	O(log n)
Worst Case	O(n)	O(log n)

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

For an e-commerce platform where speed and efficiency are crucial, binary search is generally a better choice. It significantly reduces the number of comparisons, especially when the product list is large. However, it requires that the data be sorted, which can add a one-time cost but pays off during frequent searches. Use binary search for your platform's product search—it's faster, more efficient, and scales well as your product catalog grows. Linear search can still be useful in cases where data isn't sorted or the dataset is very small.