**Solution 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.

**>> Explain Big O notation and how it helps in analyzing algorithms.**

* **Definition:** Big O notation describes the upper bound of an algorithm’s runtime complexity. It provides a way to evaluate the performance of an algorithm in terms of the size of the input data (n).
* **Purpose:** Helps to understand how the time or space requirements of an algorithm grow as the size of the input increases. It provides a way to compare the efficiency of different algorithms.

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

* **Best Case**: The search element is found immediately. For linear search, this is O(1). For binary search, this is also O(1) if the element is at the midpoint of the array.
* **Average Case**: The element is somewhere in the middle of the search process. Linear search is O(n), while binary search is O(log n).
* **Worst Case**: The element is not found or is at the end of the search process. Linear search is O(n), and binary search is O(log n) if the array is sorted.

**Linear Search**: Suitable for small datasets or unsorted arrays. It doesn’t require sorting and can handle any type of search attribute.

**Binary Search**: Suitable for large, sorted datasets. It requires sorting the array initially but provides faster search times with O(log n) complexity compared to linear search.

**Analysis**

**>> Time Complexity Comparison:**

* Linear Search:
  + Best Case: O(1) – The item is the first in the array.
  + Average Case: O(n) – The item is found on average in the middle.
  + Worst Case: O(n) – The item is not found or is at the end.
* Binary Search:
  + Best Case: O(1) – The item is in the middle of the sorted array.
  + Average Case: O(log n) – The search space is halved each time.
  + Worst Case: O(log n) – The item is not found, but the search space is reduced logarithmically.

**>> Which Algorithm is More Suitable?**

* **Linear Search** is suitable for small or unsorted datasets where simplicity is preferred, and the cost of sorting is not justifiable.
* **Binary Search** is more efficient for large, sorted datasets due to its logarithmic time complexity. However, it requires the data to be sorted beforehand, which involves additional time and space complexity for sorting.

**How to Run the code**

* Run EcommercePlatform.java file