**Data Structures and Algorithms**

Exercise 2: E-commerce Platform Search Function’

**Step1:**

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

Big Oh Notation is a way of measuring and determining how the time increases as the size increases of the input to any operation.

This notation can help us with classifying which algorithm is working the best in a scenario and thus help us choosing the right one for the operation that we wish to perform. Because in large scale applications the time taken to perform an operation matters a lot so this becomes very vital such cases.

There can be 3 types of scenarios:  
1. Best-case Scenario: In this scenario we tend to the find the element that we are looking for in the first try itself.

1. Average-case Scenario: The runtime that is based on average input. Like in our case may be finding the input in the middle of the product array.
2. Worst-case Scenario: This is the longest possible runtime of the program which may be the case when the product we are looking at is not present in the array.

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

**Linear search** checks each item one by one from start to finish. Its worst and average cases are O(n), meaning it grows directly with the number of items.

**Binary search** is much more efficient for large, sorted lists, because it dramatically reduces the number of items it needs to check by repeatedly dividing the list in half.

**Step4:**

1. **Compare the time complexity of linear and binary search algorithms**

**Linear search** is simple and works on any list, but becomes slow as the number of products grows, because in the worst case, it might have to check every single product.

**Binary search** is much more efficient for large, sorted lists, because it dramatically reduces the number of items it needs to check by repeatedly dividing the list in half.

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

In our case as we aim to achieve highest possible efficiency for our application better choice would be to use Binary Search as its time complexity is less compared to linear search because without looking at the whole array it reduces the search space to half in every iteration.

**OUTPUT:**

