**Exercise 2: E-commerce Platform Search Function**

**1. Understand Asymptotic Notation**

**1.1 Explain Big O Notation**

Big O notation describes the upper bound of the time complexity of an algorithm as a function of the input size (n). It helps in analyzing how the runtime of an algorithm grows as the input size increases. The notation focuses on the worst-case scenario, providing a way to compare the efficiency of different algorithms.

**1.2 Best, Average, and Worst-Case Scenarios for Search Operations**

* **Best Case**: The scenario where the search operation takes the least time. For example, in a linear search, the best case is when the target element is at the first position (O(1)).
* **Average Case**: The expected time complexity over all possible inputs. For linear search, the average case is O(n/2), which simplifies to O(n).
* **Worst Case**: The scenario where the search operation takes the most time. For linear search, this happens when the target element is at the last position or not present at all (O(n)). For binary search, the worst case is when the search space is halved repeatedly until one element remains (O(log n)).

**4. Analysis**

**Step 4.1: Time Complexity**

* **Linear Search**: O(n)
  + Best Case: O(1) (element is the first in the array)
  + Average Case: O(n/2) = O(n) (element is somewhere in the middle)
  + Worst Case: O(n) (element is the last or not present)
* **Binary Search**: O(log n)
  + Best Case: O(1) (element is the middle element)
  + Average Case: O(log n)
  + Worst Case: O(log n) (element requires maximum halving steps)

**Step 4.2: Suitable Algorithm for the Platform**

* **Linear Search**: Suitable for small datasets where the overhead of sorting is not justified.
* **Binary Search**: Ideal for large datasets due to its logarithmic time complexity. However, it requires the array to be sorted.