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

**1. Understand Asymptotic Notation**

**Big O Notation**

**Big O notation** is a mathematical notation used to describe the upper bound of an algorithm's time or space complexity. It gives us a way to express the worst-case scenario of an algorithm in terms of input size, helping us understand how the algorithm's runtime grows as the input size increases.

* **Purpose**: To provide a high-level understanding of an algorithm's efficiency and scalability.
* **Notation**: Denoted as O(f(n)), where f(n) describes the growth rate in relation to the input size n.

**Common Big O Notations**

1. **O(1) - Constant Time**
2. **O(log n) - Logarithmic Time**
3. **O(n) - Linear Time**
4. **O(n log n) - Linearithmic Time**
5. **O(2^n) - Exponential Time**

**Best, Average, and Worst-case Scenarios**

Understanding different scenarios helps in evaluating an algorithm’s performance under various conditions:

1. **Best Case**: The scenario where the algorithm performs the fastest.
   * Example: Finding the first element in a linear search.
2. **Average Case**: The scenario that represents the expected time complexity for typical inputs.
3. **Worst Case**: The scenario where the algorithm takes the longest time to execute.
   * Example: Searching for a non-existent element in a linear search.

**2. Analysis: LINEAR SEARCH**

* **Best Case**: O(1) - The target is the first element.
* **Average Case**: O(n/2) - On average, it checks half the elements.
* **Worst Case**: O(n) - The target is the last element or not present.

**3.Analysis: BINARY SEARCH**

* **Best Case**: O(1) - The target is the middle element.
* **Average Case**: O(log n) - The array is halved each iteration.
* **Worst Case**: O(log n) - The target is at one of the ends or not present.

**4.Which Algorithm is More Suitable?**

* **Linear Search**:
  + **Pros**: Simplicity, works on unsorted arrays, no additional setup needed.
  + **Cons**: Slow for large datasets due to O(n) complexity.
* **Binary Search**:
  + **Pros**: Fast search for large datasets due to O(log n) complexity.
  + **Cons**: Requires a sorted array, which may incur additional preprocessing time.

**Conclusion**

**Binary Search** is generally more suitable for an e-commerce platform due to its logarithmic time complexity, which ensures faster performance on large datasets.