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

**Asymptotic Notations:**

**Big O Notation:**

* **Definition:** Big O notation describes the upper bound of the time complexity of an algorithm in the worst-case scenario. It helps to understand how the runtime of an algorithm grows with the input size.
* **Purpose:** It provides a high-level understanding of the efficiency of an algorithm, allowing for comparison between different algorithms.

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

* **Best Case:** The minimum time an algorithm takes to complete. For example, in a search operation, finding the desired element in the first position.
* **Average Case:** The expected time for an algorithm to complete, averaged over all possible inputs.
* **Worst Case:** The maximum time an algorithm takes to complete. For example, searching for an element not present in the array.

**Time Complexity Analysis:**

* **Linear Search:**
  + **Best Case:** O(1). When the element is the first item in the array.
  + **Average Case:** O(n). When, on average, the element might be in the middle.
  + **Worst Case:** O(n). When the element is the last item or not present.
* **Binary Search:**
  + **Best Case:** O(1). When the element is the middle item in the sorted array.
  + **Average Case:** O(log n). When the search space is halved each time.
  + **Worst Case:** O(log n). When the element is at one end or not present.

**Comparison and Suitability:**

* **Linear Search:**
  + Suitable for small datasets or unsorted arrays.
  + Simple to implement but inefficient for large datasets.
* **Binary Search:**
  + Suitable for large, sorted datasets.
  + More efficient with O(log n) time complexity but requires the array to be sorted.
  + Sorting the array takes O(n log n) time, but once sorted, binary search is very efficient.