**Exercise 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.

**Big O Notation and Its Importance**

Big O notation helps me understand the efficiency of algorithms by describing how their runtime or space requirements grow as the input size increases. It's crucial for predicting performance and scalability.

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

* **Linear Search**:
  + **Best-case**: O(1), when the product is the first element.
  + **Average-case**: O(n), where n is the number of elements.
  + **Worst-case**: O(n), when the product is the last element or not present.
* **Binary Search**:
  + **Best-case**: O(1), if the product is in the middle.
  + **Average-case**: O(log n), assuming the array is large and sorted.
  + **Worst-case**: O(log n), when the product is not found.

**Time Complexity Comparison**

* **Linear Search**: O(n) - checks each element one by one.
* **Binary Search**: O(log n) - repeatedly halves the search space.

**Suitability for the Platform**

Binary search is more suitable for my e-commerce platform if the product list is sorted because it significantly reduces the number of comparisons needed, making it faster for large datasets. However, for small or unsorted datasets, linear search might be simpler and equally effective.