**Analysis**

**Time Complexity:**

* **Linear Search:**
  + **Best Case:** O(1) (element is at the first position).
  + **Average Case:** O(n) (element is somewhere in the middle).
  + **Worst Case:** O(n) (element is not in the array or at the end).
* **Binary Search:**
  + **Best Case:** O(1) (element is in the middle).
  + **Average Case:** O(log n) (element is somewhere in the middle, but array is halved each iteration).
  + **Worst Case:** O(log n) (element is not in the array or at one of the ends).

**Which Algorithm is More Suitable:**

* **Linear Search:** Suitable for small or unsorted datasets, as it does not require sorting.
* **Binary Search:** More efficient for large, sorted datasets due to its O(log n) time complexity. However, it requires the array to be sorted beforehand, which adds a preprocessing step.

In an e-commerce platform where search speed is critical and products are often sorted or can be sorted, **binary search** is typically more suitable due to its logarithmic time complexity. For unsorted or smaller datasets, **linear search** might be sufficient and simpler to implement.