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

**Compare the time complexity of linear and binary search algorithms**

**Linear Search:**

* **Best Case:** O(1) → Element found at the first index.
* **Average Case:** O(n) → Element is somewhere in the middle.
* **Worst Case:** O(n) → Element is at the end or not found.
* **Space Complexity:** O(1) → Constant extra space used.

**Binary Search (only works on sorted arrays):**

* **Best Case:** O(1) → Middle element is the target.
* **Average Case:** O(log n) → Continually halves the search range.
* **Worst Case:** O(log n) → Element not found after full halving.
* **Space Complexity:** O(1) for iterative, O(log n) for recursive due to call stack.

**Discuss which algorithm is more suitable for your platform and why**

For an **e-commerce platform**, where search speed is critical and datasets can be very large:

**Binary Search is more suitable, because:**

* It provides **much faster performance** (O(log n) vs. O(n)).
* Most product listings can be **pre-sorted by product ID or name**, which binary search requires.
* E-commerce search features are usually backed by **search engines or databases** that support sorted indexes, making binary search practical and efficient.

**Linear Search is only suitable:**

* When data is **small**, **unsorted**, or constantly changing.
* When search conditions are complex and don’t suit binary logic (e.g., fuzzy matching).