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

**Understanding Asymptotic Notation**

Big O Notation

Big O notation describes the upper bound of an algorithm's time or space complexity in relation to its input size. It helps us:

* Analyze how an algorithm scales with larger inputs
* Compare efficiency between different algorithms
* Predict performance characteristics

Search Operation Scenarios

* **Best-case**: The element is found immediately (O(1) for both linear and binary search)
* **Average-case**: The element is found after searching half the elements (O(n) for linear, O(log n) for binary)
* **Worst-case**: The element isn't present or is at the end (O(n) for linear, O(log n) for binary)

**Analysis and Recommendations**

Time Complexity Comparison

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

Platform Suitability

For an e-commerce platform:

1. **Binary Search** is better for:
   * Searches by product ID (unique identifier)
   * Large product catalogs where performance matters
   * Requires initial sorting (O(n log n)) but pays off with faster searches
2. **Linear Search** is better for:
   * Searches by name or category where exact matches aren't guaranteed
   * Smaller datasets where simplicity is more important than speed
   * When the data isn't sorted or changes frequently
3. **Additional Recommendations**:
   * For production systems, consider using a hash table (O(1) average case) for ID lookups
   * Implement a search index (like Elasticsearch) for complex text searches
   * Use caching for frequently accessed product

**Output**

