**E-commerce Platform Search Function Optimization**

**Understanding Asymptotic Notation (Big O)**

Big O Notation is a mathematical notation that describes the limiting behaviour of a function when the argument tends toward infinity. In computer science, it's used to classify algorithms according to how their run time or space requirements grow as the input size grows.

Search Operation Scenarios:

* **Best Case**: The element is found immediately means first element checked
* **Average Case**: The element is found after checking half the elements
* **Worst Case**: The element is not present or is the last element checked

**Analysis and Comparison**

Time Complexity:

| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** | **Space Complexity** |
| --- | --- | --- | --- | --- |
| Linear Search | O(1) | O(n) | O(n) | O(1) |
| Binary Search | O(1) | O(log n) | O(log n) | O(1) |

**Binary Search is preferable when:**

1. The product size is large (n > 1000 items)
2. You can maintain the array in sorted order (by productId)
3. Most searches are by productId (which can be sorted)
4. You can afford the O(n log n) initial sorting cost

**Linear Search is better when:**

1. The product size is small (n < 100 items)
2. You need to search by multiple fields (name, category)
3. The data changes frequently (avoiding repeated sorting)
4. You need simple implementation with no preprocessing

**Recommended Approach for E-commerce:**

1. **For productId searches (exact match):**
   * Use Binary Search on a sorted array (indexed by productId)
   * Maintain the sorted order when adding new products
2. **For productName/category searches (partial/fuzzy match):**
   * Use Linear Search (or implement a search index like Trie for names)
   * Consider using a Map<String, Product> for category searches