**Exercise 2: E-commerce Platform Search Function (Theory Answers)**

**Step 1: Understand Asymptotic Notation**

**What is Big O notation and how does it help in analyzing algorithms?**

Big O notation describes the **upper bound** of an algorithm's running time as a function of the input size n. It helps in analyzing:

* How an algorithm scales with increasing data
* Comparing performance between different algorithms
* Identifying bottlenecks and guiding optimization efforts

**Examples of common time complexities:**

* **O(n)** – Time increases linearly with input size
* **O(log n)** – Time increases slowly even as input size grows

**What are the best, average, and worst-case scenarios for search operations?**

| **Case** | **Explanation** |
| --- | --- |
| **Best Case** | Target is found immediately (e.g., first element) → fastest outcome |
| **Average Case** | Target is somewhere in the middle → realistic performance estimate |
| **Worst Case** | Target is not found or is at the last position → slowest outcome |

**Examples:**

**Linear Search:**

* Best: O(1)
* Average: O(n/2) ≈ O(n)
* Worst: O(n)

**Binary Search** *(requires sorted data)*:

* Best: O(1)
* Average: O(log n)
* Worst: O(log n)

**Step 4: Analysis**

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

| **Algorithm** | **Time Complexity** | **Requires Sorted Data** |
| --- | --- | --- |
| Linear Search | O(n) – Linear Time | No |
| Binary Search | O(log n) – Logarithmic Time | Yes |

**Which algorithm is more suitable for this e-commerce platform, and why?**

**Binary Search** is more suitable if the product data is **sorted** (e.g., by product ID or name), because it:

* Delivers significantly faster performance (O(log n)) on large datasets
* Scales well as the number of products increases
* Provides a better user experience in high-traffic e-commerce applications

However, if the data is **unsorted** and subject to frequent changes (e.g., real-time additions), **Linear Search** may be used. But this comes with a performance trade-off.

**Conclusion:**  
For optimal performance, use **Binary Search** on sorted product data. It ensures faster search results, which is critical for an e-commerce platform that handles a large number of products and users.