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

**Q. Explain Big O notation and how it helps in analyzing algorithms.**

**Ans**: Big O Notation is a mathematical notation used to describe the time or space complexity of an algorithm in terms of input size n. It gives an upper bound on the running time, helping you understand how the algorithm performs as the input grows large.

**Q. Describe the best, average, and worst-case scenarios for search operations.**

**Ans**: **a. Best Case (Ω):**

* The minimum time the algorithm takes.
* Example (Linear Search): Target element is at the firstposition.
* Time Complexity: Ω(1)

**b. Average Case (θ):**

* The expected time for a random input.
* It depends on probability distribution of input.
* Example (Linear Search): Target is somewhere in the middle.
* Time Complexity: θ(n/2) ≈ O(n)

**c. Worst Case (O):**

* The maximum time the algorithm might take.
* Example (Linear Search): Target is at the last position or not present at all.
* Time Complexity: O(n)

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

**Ans**: **Linear Search:**

* Best Case: O(1)
* Average Case: O(n)
* Worst Case: O(n)
* Requirement: Works on unsorted data; no sorting needed.

**Binary Search:**

* Best Case: O(1)
* Average Case: O(log n)
* Worst Case: O(log n)
* Requirement: Requires data to be sorted beforehand.

Linear Search checks each product one by one until it finds a match. It is simple to implement and does not require the data to be sorted. However, its performance decreases as the number of products increases because it may need to scan every element.

Binary Search works only on sorted data. It divides the dataset into halves repeatedly and narrows down the search range very quickly. This makes it significantly faster on large datasets compared to linear search.

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

**Ans**: Binary Search is more suitable for an e-commerce platform because:

* E-commerce platforms typically contain large numbers of products.
* Binary Search is more efficient for large datasets, with an average-case time complexity of O(log n).
* Product data can be sorted and indexed in advance, allowing binary search to be used effectively.
* Faster search performance enhances user experience, especially when multiple users are searching at the same time.