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

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

* Big O Notation is used to describe the performance characteristics of an algorithm, particularly its time complexity and space complexity in the worst-case scenario.
* It provides an upper bound on the time or space required by an algorithm as a function of the input size n.

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

* **Best Case:** The scenario where the algorithm performs the minimum number of operations. For search algorithms, this might be finding the target on the first try.
* **Average Case:** The expected number of operations under typical conditions. It is often more realistic and useful than the best or worst case.
* **Worst Case:** The scenario where the algorithm performs the maximum number of operations. For search algorithms, this could mean the target is not found or is found after examining all elements.

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

1. **Linear Search:**

* **Best Case:** O(1) (when the element found at the first position)
* **Average Case:** O(n) (when the element found somewhere in the array)
* **Worst Case:** O(n) (when element not found or found at the last position)

1. **Binary Search:**

* **Best Case:** O(1) (when the element found at the middle of the array)
* **Average Case:** O(log n) (when element found after several halving operations)
* **Worst Case:** O(log n) (when element not found, but all divisions done)

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

* Linear search is simple to implement and it does not require sorted data. But it is less efficient for large datasets as its worst-case time complexity is O(n).
* Binary search with the worst-case time complexity of O(log n) is more efficient but requires the data to be sorted.
* Therefore for the e-commerce platform with large datasets, binary search provides a significant performance over linear search.