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

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

Big O notation is a mathematical notation used to describe the upper bound of an algorithm's runtime or space requirements in terms of the input size. It helps in analysing the efficiency of algorithms by providing a high-level understanding of their behaviour as the input size grows. It abstracts away constants and lower-order terms to focus on the most significant factors that affect performance.

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

Best case- The scenario where the algorithm performs the fewest operations. For example, in a linear search, the best case is when the target element is the first element in the array (O(1)).

Average case- The scenario that represents the expected number of operations across all possible inputs. For linear search, this is when the target element is somewhere in the middle of the array (O(n/2), which simplifies to O(n)).

Worst case- The scenario where the algorithm performs the maximum number of operations. For linear search, the worst case is when the target element is the last element in the array or not present at all (O(n)).

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

Linear Search:

**Best Case:** O(1) - The target product is the first element.

**Average Case:** O(n) - The target product is in the middle or average position.

**Worst Case:** O(n) - The target product is the last element or not present.

Binary Search:

**Best Case:** O(1) - The target product is the middle element.

**Average Case:** O(log n) - Each step divides the array in half.

**Worst Case:** O(log n) - The array is divided in half each time until the target product is found or the search space is exhausted.

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

**Binary search** is more suitable for the e-commerce platform due to its O(log n) time complexity, which is significantly faster for large datasets compared to the O(n) time complexity of linear search. However, binary search requires the array to be sorted, which adds an initial overhead but provides much faster search times for subsequent queries. Linear search can be useful for small datasets or unsorted data, but it becomes inefficient as the dataset grows.