**E-Commerce Platform Search Function**

## **BIG NOTATION**

Big O notation is a mathematical concept used to describe the upper bound of the time complexity of an algorithm. It provides a high-level understanding of how the runtime of an algorithm scales with the size of the input data. For instance, O(n)O(n)O(n) means the runtime increases linearly with the size of the input, while O(log n)O(\log n)O(log n) means the runtime increases logarithmically.

## **PURPOSE**

It helps to evaluate and compare the efficiency of algorithms, especially when dealing with large datasets. By focusing on the most significant factors affecting performance, Big O notation abstracts away constant factors and lower-order terms to provide a clear picture of an algorithm's efficiency.

**Linear Search:**

**Best Case:** O(1) – The target element is at the very beginning of the list.

**Average Case:**  O(n) – On average, the search needs to look at half of the elements.

**Worst Case:**  O(n) – The target element is at the end of the list or not present.

**Binary Search:**

* **Best Case:** O(1) – The target element is exactly in the middle of the list
* **Average Case:** O(log n) – Each comparison halves the search space, leading to a logarithmic number of steps.
* **Worst Case:** O(log n) – Similar to the average case, the search divides the array logarithmically.

**Time Complexity Comparison:**

* **Linear Search:**

**Best Case:** O(1) – Target is found at the first position.

**Average Case:** O(n) – On average, you will check half of the elements.

**Worst Case:** O(n) – Target is at the end or not in the list.

* **Binary Search:**

**Best Case:** O(1) – Target is in the middle of the list.

**Average Case:** O(log n) – Each comparison divides the search space in half.

**Worst Case:** O(log n) – The list has to be repeatedly halved.

### **Which Algorithm is More Suitable?**

For an e-commerce platform with potentially large datasets and frequent searches, **binary search** is generally more suitable as it offers much faster search times with O(log⁡n)O(\log n)O(logn) complexity, assuming the dataset can be kept sorted or sorted as needed. If the dataset is small or unsorted, linear search might still be practical due to its simplicity.

Top of Form

Bottom of Form