**E-commerce Platform Search Function**

**Big O Notation**

* **Big O Notation** describes the *upper bound* of an algorithm’s running time.
* It helps us understand how an algorithm's runtime **grows with input size (n)**.

**Example**:

* O(1): Constant time.
* O(log n): Logarithmic - binary search.
* O(n): Linear - linear search.

**Case scenarios:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| Linear Search | **O(1)** | **O(n)** | **O(n)** |
| Binary Search | **O(1)** | **O(log n)** | **O(log n)** |

*NOTE: Setup and Implementation are found in the* ***“EcommerceSearch.java”*** *file*

**Analysis:**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Time Complexity** | **Sorting Required** |
| Linear Search | O(n) | No |
| Binary Search | O(log n) | Yes |

**Conclusion:**

**Binary search** is more suitable for an e-commerce platform because it offers significantly faster search performance (O(log n)) compared to linear search (O(n)). Since product data is usually sorted, binary search ensures quick and efficient lookups, making it ideal for large-scale applications where speed and scalability are critical.