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

1.Big O notation describes the **upper bound** of an algorithm’s runtime or space requirements as a function of the input size n. It helps determine **scalability** and performance under growth.

**O(1)**: Constant time

**O(log n)**: Logarithmic time (e.g., binary search)

**O(n)**: Linear time (e.g., linear search

**O(n log n)**: Log-linear time (e.g., merge sort)

**O(n²)**: Quadratic time (e.g., nested loops)

**Best case**: Found at first index.

**Average case**: Found somewhere in the middle.

**Worst case**: Not found or found at the end.

1. Analysis:

| **Algorithm** | **Time Complexity** | **Use Case** |
| --- | --- | --- |
| **Linear Search** | O(n) | Small/unsorted datasets or simple filters |
| **Binary Search** | O(log n) | Large/sorted datasets, real-time performance critical |

### ****Which is better?****

For **real-time search** with many products (e.g., 1M+), **binary search is far superior**.

However, it **requires sorted data**, which adds a one-time cost: O(n log n) for sorting.

In practice, **modern platforms index products by multiple attributes** using advanced data structures or databases (e.g., B-trees, hash maps, search engines like Elasticsearch).