## **Exercise 2: E-Commerce Platform Search Function**

## **What is Big O Notation?**

**Big O notation** is a way to describe the **time or space complexity of an algorithm in terms of input size (n)**. It tells us **how the performance of an algorithm changes when the size of the input increases**.

In simple words, Big O notation helps us understand:

* How much time an algorithm will take
* How much memory it will use
* How it behaves as data size grows larger

## **Why is it Important**

* It allows us to **compare different algorithms** based on their efficiency.
* It helps identify which algorithm is better for **small or large datasets**.
* It focuses on the **growth rate**, not exact execution time, which makes it reliable across different systems.
* Great idea, Mohit — here’s your **best, average, and worst-case explanation in a clean, easy-to-read table format**:

## **Search Operation Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **When it Happens** | **Linear Search** | **Binary Search** |
| **Best Case** | Item is found in the **first try** | O(1) | O(1) |
| **Average Case** | Item is found **somewhere in the middle** | O(n) | O(log n) |
| **Worst Case** | Item is at the **last position** or **not found** | O(n) | O(log n) |

* **Linear Search**: Slower, checks one by one
* **Binary Search**: Faster, divides list in half each time (needs sorted list)

