# **Understand Asymptotic Notation:**

Big O notation describes the upper bound of an algorithm's running time or space usage as a function of input size n

| **Big O** | **Name** | **Example** |
| --- | --- | --- |
| O(1) | Constant time | Accessing array element by index |
| O(n) | Linear time | Linear search |
| O(log n) | Logarithmic time | Binary search |
| O(n²) | Quadratic time | Bubble sort |

# **Best, Average, and Worst-Case in Search**

| Search Type | Best Case | Average Case | Worst Case |
| --- | --- | --- | --- |
| Linear Search | O(1) – item at index 0 | O(n/2) = O(n) | O(n) – item not present |
| Binary Search | O(1) – middle match | O(log n) | O(log n) – item not present |

# **Analysis:**

| **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) |

**Binary Search is clearly better** **if**

* Your product list is **already sorted or rarely updated**.
* You need **frequent and fast lookups** (like search bar queries).

**Linear Search is better** **if**

* The list is **unsorted and very dynamic**.
* You're working with **small datasets**, or doing **one-off scans** (e.g., filtering by category).

CODE SOLUTION:

