**Understanding Asymptotic Notation – E-commerce Search**

Asymptotic notation helps us understand how an algorithm performs as the input size grows. It doesn't measure exact time but shows how fast or slow an algorithm scales. This is important in an e-commerce platform where the product list can grow from hundreds to thousands.

**Big O Notation** is used to describe the efficiency:

* **O(1)** – Constant time: Fast and doesn't depend on input size.
* **O(log n)** – Logarithmic time: Very efficient for large data.
* **O(n)** – Linear time: Slower as input size increases.
* **O(n²)** – Quadratic time: Gets very slow with large inputs.

In our e-commerce search engine, we used two search types:

**Linear Search**:

* Best case: Product is first.
* Average case: Product is somewhere in the middle
* Worst case: Product is last or not found
* Time Complexity: O(1), O(n), O(n)

**Binary Search** (on sorted list):

* Best case: Product is in the middle
* Average case: Found after a few comparisons
* Worst case: Not found
* Time Complexity: O(1), O(log n), O(log n)

**Which is better?**  
Use **Linear Search** for small or unsorted lists and partial keyword matches (like searching "pan" to get "frying pan").  
Use **Binary Search** for large, sorted lists when an exact match is needed (like finding "Electric Kettle").