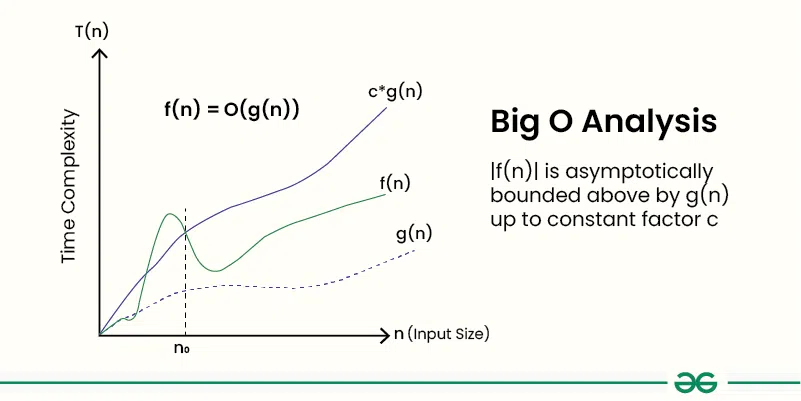
**Understanding Asymptotic Notation**

****Big-O Notation****

**Big O notation describes the upper bound of an algorithm’s** time or space complexity**. It is a powerful tool used in computer science to describe the time complexity or space complexity of algorithms.**

**It’s denoted as **O(f(n))**, where **f(n)** is a function that represents the number of operations (steps) that an algorithm performs to solve a problem of size **n**.**



**It focuses on the worst-case scenario, we can ensure performance stays acceptable even at scale.This is extremely crucial** real-world apps like e-commerce, where latency or delays during search could make potential customers dissatisfied.

****Different Scenarios For Search Operations****

**The best-case scenario occurs when the target element is found immediately. The average-case reflects typical performance. The worst-case happens when the element is at the end or not present at all, leading to the maximum number of comparisons**

| Algorithm | Best Case | Average Case | Worst Case | Big O Class |
| --- | --- | --- | --- | --- |
| **Linear** | O(1) | O(n/2) | O(n) | **O(n)** |
| **Binary** | O(1) | O(log n) | O(log n) | **O(log n)** |

**Analysis and Conclusion**

For **small or frequently updated data**, use **Linear Search** for simplicity.  
For **large, sorted, or rarely-changing datasets**, **Binary Search** is optimal due to its logarithmic performance.