Big O Notation by Vaibhav Jain (2228077)

Big O notation describes the *upper bound* of an algorithm's running time or space complexity in terms of input size.

Notation	Meaning	Example
O(1)	Constant time	Access array index
O(n)	Linear time	Linear search
O(log n)	Logarithmic time	Binary search
O(n²)	Quadratic time	Nested loops

Comparison of Time Complexity

Algorithm	Best Case	Average Case	Worst Case	Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)

Linear Search:

- Scan each element one by one.
- Takes **longer** as the dataset size increases.
- Does **not** require the array to be sorted.

Binary Search:

- Repeatedly divides the sorted array in half.
- Much **faster** than linear search for large datasets.
- Requires the array to be **sorted** beforehand.

Binary Search is more suitable for your e-commerce platform

1. **Large Product Catalog**:
    - E-commerce platforms typically have thousands or millions of items.
    - Binary search scales efficiently due to **O(log n)** performance.

2. **Sorted Product Lists**:
    - Products are usually displayed or stored in sorted order (by name, price, etc.).
    - This satisfies the **precondition** for binary search.
3. **Fast Response Requirement**:
    - Customers expect instant search results.
    - Binary search is much faster for lookups in sorted data.
4. **Minimal Memory Use**:
    - Both algorithms use O(1) space, but **binary search gives better performance** without needing extra memory.