**1. Understand Asymptotic Notation**

**Big O Notation**

Big O notation is a mathematical concept used to describe the efficiency of an algorithm by providing an upper limit on its running time or space requirements relative to the input size. It is particularly useful for comparing different algorithms by evaluating their performance in the worst-case scenario. This notation helps to understand how an algorithm behaves as the input size grows. By focusing on the worst-case scenario.

. For example:

* **O(1)**: Constant time, regardless of input size.
* **O(n)**: Linear time, proportional to input size.
* **O(log n)**: Logarithmic time, grows slower than linear time.

**Search Operation Scenarios**

* **Best Case**: The search term is found at the first element (O(1) for linear search).
* **Average Case**: The search term is found somewhere in the middle of the list (O(n) for linear search, O(log n) for binary search).
* **Worst Case**: The search term is either not found or is at the end of the list (O(n) for linear search, O(log n) for binary search).

**4. Analysis**

**Time Complexity Comparison**

* **Linear Search**:
  + **Time Complexity**: O(n)
  + **Description**: Observes each element until the desired item is found or the end of the list is reached.
* **Binary Search**:
  + **Time Complexity**: O(log n)
  + **Description**: Repeatedly divides the search interval in half, applicable to sorted data.

**Suitability for the Platform**

* **Binary Search** is preferred for the e-commerce platform if the product list is sorted, as it provides faster search performance (O(log n) versus O(n) for linear search). Linear search is suitable for unsorted lists or smaller datasets. Binary search is an efficient algorithm for finding an item in a sorted array. It works by repeatedly dividing the search interval in half. If the target value is less than the middle element, the search continues in the lower half; otherwise, it continues in the upper half.