**Week-I**

**Algorithms & DSA**

**Exercise-2:**

**Understanding Asymptotic Notation:**

**Big O Notation:**Big O notation is a mathematical notation used to describe the upper bound of an algorithm's runtime or space requirements in terms of the input size. It helps in analysing the performance of algorithms by providing a high-level understanding of their efficiency and scalability. It expresses the worst-case scenario but can also describe average and best-case scenarios.

* Best Case: The scenario where the algorithm performs the minimum number of steps.
* Average Case: The expected number of steps for a random input.
* Worst Case: The scenario where the algorithm performs the maximum number of steps.

**Time Complexity Comparison:**

* **Linear Search:**
  + Best Case: O(1) (If element is the first one in the array).
  + Average Case: O(n)
  + Worst Case: O(n) (if element is the last one)
* **Binary Search:**
  + Best Case: O(1) (element is the middle one)
  + Average Case: O(log n).
  + Worst Case: O(log n) (when the element is not found).

Binary search is more suitable for the platform because it has a significantly lower time complexity compared to linear search for large datasets. However, it requires the array to be sorted, which adds an overhead. If the product list is dynamic and frequently changing, maintaining the sorted order can be costly. In such cases, a balanced binary search tree (e.g., AVL tree or Red-Black tree) might be more suitable to maintain sorted order while supporting efficient insertions and deletions.