Ex 2: E-commerce Platform Search Function

Big O notation describes the **upper bound** of an algorithm's runtime. It shows how performance scales with the size of the input (**n**). It helps in choosing the most efficient algorithm.

- ❖ **Linear Search**
  - ➢ **Best Case:**

    O(1) – The target is found at the first position.
  - ➢ **Average Case:**

    O(n/2) ≈ O(n) – On average, it checks half the elements.
  - ➢ **Worst Case:**
    - ♦ O(n) – The target is at the last position or not present at all.
- ❖ **Binary Search**
  - ➢ **Best Case:**

    O(1) – The target is found at the middle position.
  - ➢ **Average Case**:

    O(log n) – Efficiently narrows down the search space by half each time.
  - ➢ **Worst Case:**

    O(log n) – Maximum number of divisions needed to find (or not find) the target.

Time Complexity:

Linear Search:

- Time Complexity: O(n)
- Suitable For : Small or unsorted product lists

Binary Search:

- Time Complexity : O(log n)
- Suitable For : Large, sorted product lists

Suitable Algorithm:

- For **large e-commerce platforms**, binary search (or better, hash-based or trie-based search) is preferred for speed and scalability.
- If products are not sorted, use **linear search** or **consider using HashMap** for constant-time lookup.