**E-commerce Platform Search Function**

**Understand Asymptotic Notation**

**Big O Notation**

Big O notation is used to describe the performance or complexity of an algorithm. It provides an upper bound on the time (or space) required for an algorithm to complete as a function of the input size (n). It helps in analyzing algorithms by:

* **Providing a high-level understanding**: Ignoring constant factors and lower-order terms to focus on the growth rate of the runtime or space usage as input size increases.
* **Comparing algorithms**: Allowing comparison of the efficiency of different algorithms in terms of scalability.

**Best, Average, and Worst-Case Scenarios**

* **Best Case**: The minimum time required for an algorithm to complete, typically when the desired element is the first one examined.
* **Average Case**: The expected time required for an algorithm to complete, averaged over all possible inputs.
* **Worst Case**: The maximum time required for an algorithm to complete, typically when the desired element is the last one examined or not present at all.

**Analysis**

**Time Complexity Comparison**

* **Linear Search**:
  + **Best Case**: O(1) - The product is the first element.
  + **Average Case**: O(n) - On average, the search goes through half of the array.
  + **Worst Case**: O(n) - The product is the last element or not present.
* **Binary Search**:
  + **Best Case**: O(1) - The product is the middle element.
  + **Average Case**: O(log n) - The search divides the array in half each step.
  + **Worst Case**: O(log n) - The search goes through log n levels of division.

**Suitability for the Platform**

* **Linear Search**:
  + Suitable for small datasets or unsorted data.
  + Simple to implement but inefficient for large datasets due to O(n) complexity.
* **Binary Search**:
  + Suitable for large, sorted datasets.
  + More efficient due to O(log n) complexity, but requires the data to be sorted beforehand.