**E-COMMERCE PLATFORM SEARCH FUNCTION**

1.Understand the Problem:

* Explain Big O notation and how it helps in analyzing algorithms.

Big O notation is used to describe the upper bound of an algorithm's time or space complexity. It gives an estimate of how long an algorithm takes to complete as the input size increases. Big O notation is usually expressed as a function of the input size, typically represented as 'n'.

For example,we can say, if an algorithm has a time complexity of O(n), it means the algorithm's execution time grows linearly with the size of the input.

* Describe the best, average, and worst-case scenarios for search operations.
  + **Best-Case Scenario:** The minimum time an algorithm takes to complete, usually when the input is already sorted or has a specific structure.
  + **Average-Case Scenario:** The expected scenario where the algorithm performs an average number of operations, considering all possible inputs.
  + **Worst-Case Scenario**: The maximum time an algorithm takes to complete, usually when the input is in the worst possible order, we can say the last searching place.

4. Analysis:

* Compare the time complexity of linear and binary search algorithms.

**Linear Search:**

* Best Case: O(1) (when the product is found at the first position)
* Average Case: O(n)
* Worst Case: O(n) (when the product is found at the last position or not found )

**Binary Search:**

* Best Case: O(1) (when the product is found at the middle position which is first search point)
* Average Case: O(log n)
* Worst Case: O(log n) (when the product is not found after several divisions)
* Discuss which algorithm is more suitable for your platform and why.

For an e-commerce platform where the product list can be large and the search operation is frequent, **binary search** is more suitable due to its time complexity and faster performance. However, this requires maintaining a sorted list of products, which may involve additional overhead for insertion and deletion operations.