E-COMMERCE PLATFORM

**1. Understanding Asymptotic Notation**

**Big O Notation**: Big O notation is a mathematical representation used to describe the efficiency of an algorithm. It measures the worst-case time complexity, giving an upper bound on the time an algorithm will take to run as a function of the input size (n). This helps developers understand the scalability and performance of algorithms.

* **O(1)**: Constant time complexity - the algorithm's runtime is constant regardless of input size.
* **O(log n)**: Logarithmic time complexity - the algorithm's runtime grows logarithmically with input size.
* **O(n)**: Linear time complexity - the algorithm's runtime grows linearly with input size.
* **O(n log n)**: Linearithmic time complexity - the algorithm's runtime grows in proportion to n log n.
* **O(n^2)**: Quadratic time complexity - the algorithm's runtime grows quadratically with input size.
* **O(2^n)**: Exponential time complexity - the algorithm's runtime grows exponentially with input size.

**Best, Average, and Worst-Case Scenarios for Search Operations**:

* **Best-Case Scenario**: The search finds the target on the first try, resulting in minimal operations (e.g., O(1) for the first element in a linear search).
* **Average-Case Scenario**: The search performs a typical number of operations, providing a balanced view of its performance (e.g., O(n/2) for linear search, which simplifies to O(n)).
* **Worst-Case Scenario**: The search performs the maximum number of operations, giving the upper limit of its complexity (e.g., O(n) for linear search when the element is at the end or not present).