

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

Big O notation is a mathematical notation used to describe the time complexity or space complexity of an algorithm. It provides an upper bound on the time taken by an algorithm in terms of the size of the input. This helps in analyzing and comparing the efficiency of different algorithms, especially for large input sizes.

## Describe the best, average, and worst-case scenarios for search operations.

- **Best Case:** The scenario where the algorithm performs the minimum number of operations. For example, in a search operation, the best case occurs when the target element is found at the first position.
- **Average Case:** The scenario that represents the expected number of operations for a typical input. It is often calculated by averaging the time complexities for all possible inputs.
- **Worst Case:** The scenario where the algorithm performs the maximum number of operations. For example, in a search operation, the worst case occurs when the target element is not present in the array, requiring a full scan.

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

- **Linear Search:**  $O(n)$
- **Binary Search:**  $O(\log n)$

## Discuss which algorithm is more suitable for your platform and why.

For an e-commerce platform, binary search is more suitable due to its logarithmic time complexity, which ensures faster search operations for large datasets. However, it requires the data to be sorted. If the data is not sorted or the dataset is small, linear search might be more practical.