1. **Big O Notation:**
   * **Definition**: Big O notation is a mathematical notation used to describe the upper bound (worst-case) of the time complexity of an algorithm in terms of the input size.
   * **Purpose**:
     + Helps analyse and compare the efficiency of algorithms.
     + Provides a way to express how an algorithm’s runtime grows relative to the input size.
   * **Example**:
     + If an algorithm has a time complexity of O(n), it means that the runtime grows linearly with the input size (e.g., if the input doubles, the runtime also doubles).
2. **Search Operation Scenarios:**
   * **Best Case**:
     + The most favourable scenario.
     + For search algorithms, this is when the target element is found immediately (e.g., binary search on a sorted array).
     + Time complexity: O(1) (constant time).
   * **Average Case**:
     + Represents the expected performance over a range of inputs.
     + Often harder to analyse precisely.
     + For search algorithms, this considers the distribution of data and the likelihood of finding the target.
   * **Worst Case**:
     + The scenario where the algorithm performs the slowest.
     + For search, this is when the target element is at the last position or not present at all.
     + Time complexity: Depends on the algorithm (e.g., linear search: O(n), binary search: O(log n)).