**EX 2**

**Understanding Asymptotic Notation**

**Big O Notation**

* **Definition**: Big O notation describes the upper bound of an algorithm's running time, providing an estimate of the worst-case scenario in terms of time complexity. It helps in understanding how an algorithm scales with the input size.
* **Purpose**: It abstracts away constants and lower-order terms, focusing on the dominant factors that affect performance as input size grows.

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

* **Best-case**: The scenario where the algorithm performs the minimum number of operations. For search operations, this could be finding the target element on the first try.
* **Average-case**: The scenario that represents the expected performance of the algorithm over all possible inputs. It gives a realistic expectation of the algorithm's performance.
* **Worst-case**: The scenario where the algorithm performs the maximum number of operations. For search operations, this could be not finding the target element until the end of the search.

**Analysis**

**Time Complexity**

1. **Linear Search**:
   * **Best-case**: O(1) - The element is found at the first position.
   * **Average-case**: O(n) - The element is in the middle.
   * **Worst-case**: O(n) - The element is at the last position or not found at all.
2. **Binary Search**:
   * **Best-case**: O(1) - The element is at the middle position.
   * **Average-case**: O(log n) - Each comparison's search space is halved.
   * **Worst-case**: O(log n) - The element is not found after log n divisions.
   * Note: Binary search requires the array to be sorted, which may add an O(n logn) complexity if the array isn't already sorted.

**Suitable Algorithm for the Platform**

* **Linear Search**:
  + Pros: Simple to implement, works on unsorted arrays.
  + Cons: Inefficient for large datasets as it has a time complexity of O(n)O(n)O(n).
* **Binary Search**:
  + Pros: Highly efficient with a time complexity of O(log⁡n)O(\log n)O(logn) for sorted arrays.
  + Cons: Requires the array to be sorted, which may add an overhead if the array needs frequent sorting.