**WEEK1\_ALGORITHMS\_DATA\_STRUCTURES**

**EXERCISE-2**

**Understand Asymptotic Notation**

**Big O Notation:**

Big O notation is a mathematical notation used to describe the upper bound of an algorithm's running time or space complexity. It helps in analyzing the performance and efficiency of algorithms, particularly in terms of how they scale with the size of the input.

* **O(1):** Constant time – the operation's time is independent of the input size.
* **O(log n)**: Logarithmic time – the operation's time grows logarithmically with the input size.
* **O(n)**: Linear time – the operation's time grows linearly with the input size.
* **O(n log n)**: Linearithmic time – the operation's time grows in proportion to n log n.
* **O(n^2)**: Quadratic time – the operation's time grows quadratically with the input size.

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

**Best Case**: The minimum time required for an algorithm to complete. For search operations, it often occurs when the target element is at the beginning of the dataset.

**Average Case**: The expected time required for an algorithm to complete, considering all possible inputs.

**Worst Case**: The maximum time required for an algorithm to complete. For search operations, it usually occurs when the target element is at the end of the dataset or not present at all.

**Analysis**

**Time Complexity Comparison**

**Linear Search:**

* **Best Case: O(1)** – when the target element is the first element.
* **Average Case: O(n)** – the target element is somewhere in the middle.
* **Worst Case: O(n)** – the target element is at the end or not present at all.

**Binary Search:**

* **Best Case: O(1)** – when the target element is the middle element.
* **Average Case: O(log n)** – the search space halves with each comparison.
* **Worst Case: O(log n)** – the search space continues to halve until the element is found or not present.

**Suitable Algorithm for E-commerce Platform**

**Linear Search:**

* Simple to implement and does not require sorted data.
* Suitable for small datasets or unsorted data.

**Binary Search:**

* Requires sorted data but provides significantly faster search times for large datasets.
* More suitable for e-commerce platforms with large inventories due to its O(log n) time complexity for search operations.