**Asymptotic Notation**

**Big O Notation**

* Big O notation: It's a mathematical notation used to describe the upper bound of an algorithm's running time. It provides an estimate of the maximum time an algorithm will take to run as a function of the input size (n).

Examples: 0(n), O(log n), O(n2), etc

Best, Average, and Worst-Case Scenarios

**Best Case**: The scenario where the algorithm performs the minimum number of steps.

* For linear search: O(1) (if the target is the first element).
* For binary search: O(1) (if the target is the middle element).

**Average Case**: The expected scenario where the algorithm performs an average number of steps.

* For linear search: O(n/2) ≈ O(n) .
* For binary search: O(log n).

**Worst Case**: The scenario where the algorithm performs the maximum number of steps.

* For linear search: O(n).
* For binary search: O(log n).

**Analysis**

**Time Complexity Comparison**

* **Linear Search:**
  + Best Case: O(1)
  + Average Case: O(n)
  + Worst Case: O(n)
* **Binary Search:**
  + Best Case: O(1)
  + Average Case: O(log n)
  + Worst Case: O(log n)

**Suitability for the Platform**

* **Linear Search:** Suitable for small datasets or unsorted datasets. It's simple to implement but not efficient for large datasets.
* **Binary Search:** Suitable for large, sorted datasets. It is much faster than linear search for large datasets due to its logarithmic time complexity.