Understand Asymptotic Notation

Big O Notation:

Big O Notation is a mathematical notation used to describe time or space complexity of computer programs. It is a way to express the upper bound of an algorithm's time or space complexity.

It describes the speed of the growth of the algorithm like how fast or how slow the algorithms performs as per input size. It doesn't directly give the exact value.

It helps to calculate the time and space complexity of computer programs using the above process.

Best, Average, Worst Case Scenarios for Search Algorithms:

Linear Search: Useful in sorted and unsorted arrays as well.

Best Case: O(1) -> When the answer is at starting position (index =0)

Average Case: O(n)

Worst Case: O(n) -> When the answer is at the end or the answer is not there in the array.

Binary Search: Used only on sorted array!

Best Case: O(1) -> When the answer is at the middle (index = n/2)

Average Case: O(log n)

Worst Case: O(log n) -> When answer is not found or is at one end.

Analysis

Time Complexity of Linear Search & Binary Search:

Time Complexity of Linear Search: O(n)

No need of sorted array. Can be done even on unsorted arrays. Better for small arrays only… As the size increase time increases.

Time Complexity of Binary Search: O(log n)

This can be performed only on sorted arrays only, cannot be done on unsorted arrays. Better when the size of input array increases…

Which algorithms if best for this scenario?:

Binary Search algorithm is best for this use case because the product list can go to lakhs and crores. In that scenario binary search is helpful as it offers lower search time for larger arrays.