**1.Understand Asymptotic Notation:**

Big O notation is a very useful tool used in Computer Science. It describes time or space complexity of algorithms. It expresses the upper bound of an algorithm’s space or time complexity.

Given two functions**f(n)** and **g(n)**, we say that**f(n)** is**O(g(n))** if there exist constants**c > 0** and **n0 >=**0 such that**f(n) <= c\*g(n)** for all **n >= n0**.

Features:

1. It describes the asymptotic behavior of a function i.e. function’s order of growth or time depending on the input size.
2. It can be used to compare the efficiency of different algorithms or data structures.
3. It considers the worst case scenario of the algorithm to find its time complexity in terms of Big O.
4. It is denoted as **O(f(n))**, where **f(n)** is a function that represents the number of steps an algorithm performs to solve a problem of size **n**.

Common Big(0):

1. O(1): Constant Time
2. O(n): Linear Time
3. O(log n): Logarithmic Time
4. O(n2): Quadratic Time

Best, Average and Worst Case for Search:

|  |  |  |  |
| --- | --- | --- | --- |
| **ALGORITHM** | **BEST CASE** | **AVERAGE CASE** | **WORST CASE** |
| Linear Search | O(1) | O(n/2)🡪O(n) | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |

**4. Analysis:**

Time Complexity Comparison:

|  |  |  |
| --- | --- | --- |
| **ALGORITHM** | **TIME COMPLEXITY** | **SORTED REQUIRED** |
| Linear Search | O(n) | No |
| Binary Search | O(n) | Yes |

Preferable Algorithm:

Linear Search is preferred for Small Datasets. It is simpler and is more flexible.

Whereas, Binary Search is preferred for Large Datasets. It is faster and is more scalable than Linear Search.

Since my platform size is small, Linear Search is preferred for implementing a simple and flexible logic for searching.