**need of data structures and algorithms:**

Data structures allow us to organize and store data, while algorithms allow us to process that data in a meaningful way.

**Time complexity :**

Time complexity is a function that describes how long an algorithm takes in terms of the quantity of input it receives.

**Space complexity:**

Space complexity is a function that describes how much memory (space) an algorithm requires to the quantity of input to the method.

**Asymptotic Notation :**

Asymptotic Notation is used to describe the running time of an algorithm - how much time an algorithm takes with a given input, n. There are three different notations: **big O, big Theta (Θ), and big Omega (Ω)**.

**Big-O notation:**

Big-O notation represents the upper bound of the running time of an algorithm. Thus, it gives the worst-case complexity of an algorithm.

## **Omega Notation (Ω-notation):**

## Omega notation represents the lower bound of the running time of an algorithm. Thus, it provides the best case complexity of an algorithm.

**Average and Worst case analysis:**

In the worst-case analysis, we calculate the upper bound on the running time of an algorithm. We must know the case that causes a maximum number of operations to be executed. For Linear Search, the worst case happens when the element to be searched (x) is not present in the array. When x is not present, the search() function compares it with all the elements of arr[] one by one. Therefore, the worst-case time complexity of the linear search would be O(n).

**worst case analysis:**

we guarantee an upper bound on the running time of an algorithm which is good information.

### **Average Case Analysis:**

The average case analysis is not easy to do in most practical cases and it is rarely done. In the average case analysis, we must know (or predict) the mathematical distribution of all possible inputs.

### Best Case Analysis:

The Best Case analysis is bogus. Guaranteeing a lower bound on an algorithm doesn’t provide any information as in the worst case, an algorithm may take years to run.

e.g In linear search program in array

Time Complexity Analysis: (In Big-O notation)

* Best Case:O(1),This will take place if the element to be searched is on the first index of the given list. So, the number of comparisons, in this case, is 1.
* Average Case:O(n)

This will take place if the element to be searched is on the middle index of the given list.

* Worst Case:O(n),This will take place if:
  + The element to be searched is on the last index
  + The element to be searched is not present on the list