

searching, linear search and Binary search.

- 5) **Sorting** :- The process of arranging the data structure in a specific order is called as sorting. There are many algorithms that can be used to perform sorting, for example, insertion sort, selection sort, bubble sort etc.
- 6) **Merging** :- When two lists list A and list B of size  $m$  and  $n$  respectively, of similar type of elements, clubbed or joined to produce third list, list C of size  $(m+n)$ , then this process is called merging.

## DATA STRUCTURES AND ALGORITHM

**What is Algorithm?**

An **algorithm** is a process or a set of rules required to perform calculations or some other problem-solving operations especially by a computer.

It is not complete program or code; it is just a solution (logic) of a problem, which can be represented either as an informal description using a flowchart or pseudocode.

characteristics of an **algorithm**.

**Input** :- An algorithm has some input values. We can pass 0 or some input value to an algorithm.



Output :- We will get 1 | more output at end of an algorithm.

Unambiguity :- An algorithm should be unambiguous which means that instruction in an algorithm should be clear and simple.

Finiteness :- An algorithm should have finiteness means limited number of instructions.

Effectiveness :- An algorithm should have finite as each instruction in an algorithm affects the overall process.

### Approaches in Algorithm :-

- 1) Brute force Algorithm :- The general logic structure is applied to design an algorithm. It is also known as exhaustive search algorithm that searches all possible to provide required solution.

Such algorithms have two types :-

#### 1) Optimizing

Finding all solutions of a problem and then take out the best solution is known then it will terminate if the best solution is known.

#### 2) Sacrificing

As soon as the best solution is found, then it will stop.

**Divide and conquer :-** This breaks down the algorithm to solve the problem in different methods. It allows you to break down problem into different methods, and valid output is produced for the valid input. This valid output is passed to some other function.

**Greedy algorithm :-** It is an algorithm paradigm that makes an optimal choice on each iteration with the hope of getting best solution. It is easy to implement and has faster execution time. But there are very rare cases in which it provides the optimal solution.

**The major categories of algorithms** are given below:

**Sort :-** Algorithm developed for sorting the items in a certain order.

**search :-** Algorithm developed for searching the items inside a data structure.

**Delete :-** Algorithm developed for deleting the existing element from the data structure.

**Insert :-** Algorithm developed for inserting an item inside a data structure.

**Update :-** Algorithm developed for updating the existing element inside a data structure.



## Algorithm Analysis :-

The algorithm can be analyzed in two levels i.e. first is before creating the algorithm, and second is after creating the algorithm.

There are two analysis of an algorithm.

### Priori Analysis :-

Here, priori analysis is the theoretical analysis of an algorithm which is done before implementing the algorithm.

### Posterior Analysis :-

Here, posterior analysis is a practical analysis of an algorithm. The practical analysis is achieved by implementing algorithm using any programming language.

## Algorithm complexity :-

The performance of the algorithm can be measured in two factors :

### Time complexity :-

The time complexity of an algorithm is the amount of time required to complete the execution. The time complexity of an algorithm is denoted by the big O notation.

Here big O notation is the asymptotic notation to represent time complexity. The time complexity is mainly calculated by counting the number of steps to finish execution.

```
sum = 0 ;
```

```
// Suppose we have to calculate the sum of n numbers.
```

```
for i = 1 to n
```

```
sum = sum + i ;
```

```
// When the loop ends then sum holds the sum of n numbers.
```

```
return sum ;
```

In above code, the time complexity of the loop statement will be at least  $n$ , and if value of  $n$  increases, then time complexity also increases.

We generally consider the worst-time complexity as it is maximum time taken for any given input size.

**Space complexity :-**

An algorithm's space complexity is the amount of space required to solve a problem and produce an output. Similar to the time complexity, space complexity is also expressed in big  $O$  notation.

Space complexity = Auxiliary space + Input size.



The following are the types of algorithms :

Search Algorithm :-

on each day, we search for something in our day to day life.

similarly, with the case of computer, huge data is stored in a computer that whenever user asks for any data then the computer searches for that data in the memory and provides that data to the user. There are mainly two techniques available to search data in an array :

- Linear search
- Binary search

Sorting Algorithms :-

sorting algorithms are used to rearrange elements in an array or a given data structure either in an ascending or descending order. The comparison operator decides the new order of the elements :