Data Structures & Algorithms - Overview

What is Data Structure?

Data Structure is a systematic way to organize data in order to use it efficiently. Following terms are the foundation terms of a data structure.

- **Interface** − Each data structure has an interface. Interface represents the set of operations that a data structure supports. An interface only provides the list of supported operations, type of parameters they can accept and return type of these operations.
- **Implementation** − Implementation provides the internal representation of a data structure. Implementation also provides the definition of the algorithms used in the operations of the data structure.

Types of Data Structures

Here are different type of data structures which we are going to learn in this tutorial:

- Array Data Structure
- String Data Structure
- Linked List Data Structure
- Double Linked List Data Structure
- Circular Linked List Data Structure
- Stack Data Structure
- Queue Data Structure
- Heap Data Structure
- Hash Data Structure

- Matrix/Grid Data Structure
- Graph Data Structure
- Tree Data Structure

What is Algorithm?

Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language.

Types of Algorithms

Here are different type of algorithms which we are going to learn in this tutorial:

- DSA - Searching Algorithms
- DSA - Sorting Algorithms
- DSA - Approximation Algorithms
- DSA - Divide and Conquer Algorithms
- DSA - Greedy Algorithms
- DSA - Recursion Algorithm
- DSA - Backtracking Algorithm
- DSA - Randomized Algorithms
- DSA - Dynamic Programming
- DSA - Pattern Searching
- DSA - Mathematical Algorithms
- DSA - Geometric Algorithms
- DSA - Bitwise Algorithms
- DSA - Branch and Bound Algorithm

Characteristics of a Data Structure

- **Correctness** − Data structure implementation should implement its interface correctly.
- **Time Complexity** − Running time or the execution time of operations of data structure must be as small as possible.
- **Space Complexity** − Memory usage of a data structure operation should be as little as possible.

Execution Time Cases

There are three cases which are usually used to compare various data structure's execution time in a relative manner.

- **Worst Case** − This is the scenario where a particular data structure operation takes maximum time it can take. If an operation's worst case time is ƒ(n) then this operation will not take more than ƒ(n) time where ƒ(n) represents function of n.
- **Average Case** − This is the scenario depicting the average execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then m operations will take mƒ(n) time.
- **Best Case** − This is the scenario depicting the least possible execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then the actual operation may take time as the random number which would be maximum as ƒ(n).

Basic DSA Terminologies

- **Data** − Data are values or set of values.
- **Data Item** − Data item refers to single unit of values.
- **Group Items** − Data items that are divided into sub items are called as Group Items.
- **Elementary Items** − Data items that cannot be divided are called as Elementary Items.

- **Attribute and Entity** − An entity is that which contains certain attributes or properties, which may be assigned values.
- **Entity Set** − Entities of similar attributes form an entity set.
- **Field** − Field is a single elementary unit of information representing an attribute of an entity.
- **Record** − Record is a collection of field values of a given entity.
- **File** − File is a collection of records of the entities in a given entity set.