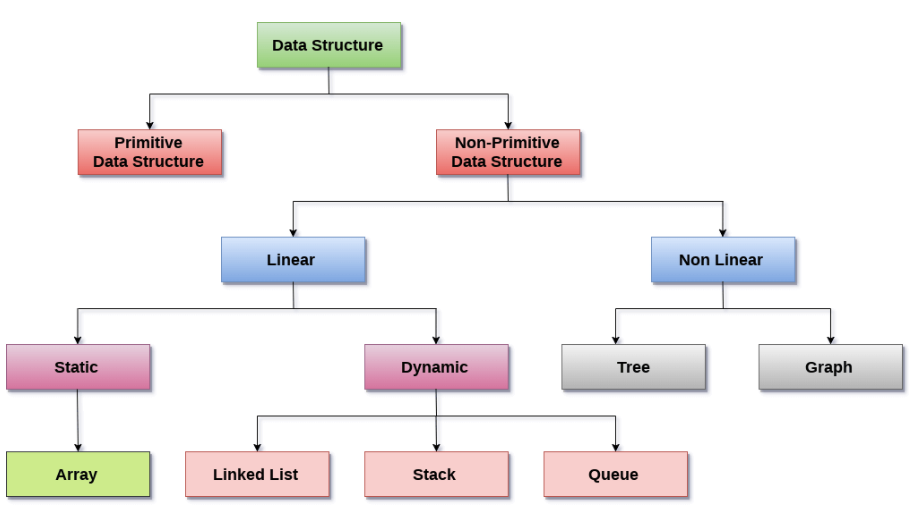
**DATA STRUCTURES WITH C**

* **Introduction**

Data Structures, as the name suggests, is a way of organizing data in memory.

It is a set of algorithms that we can use in any programming language to structure the data in the memory.

* **Types of Data Structures**



1. **Primitive Data Structures:**

Data structures that hold a single value like int, char, double, float and pointer are called primitive data structures.

1. **Non-Primitive Data Structure:**

These are further divided into two types:

1. Linear Data Structure:

The arrangement of data in a sequential manner is known as a linear data structure. One element is connected to only one another element in a linear form. Examples- Linked list, Stacks and Queues.

1. Non-Linear Data Structures:

It is when one element is connected to ‘n’ number of elements. Elements are arranged in a random manner. Example- Trees and Graphs.

Data Structures can also be classified as:

a)Static Data Structure:

Size is allocated at the compile time. Maximum size is fixed.

b)Dynamic Data Structure:

Size is allocated at the run time. Maximum size is flexible.

* **Major Operations:**

1. **Searching**
2. **Sorting**
3. **Insertion**
4. **Updation**
5. **Deletion**
6. **Traversing** (Visiting each element of the data structure in order to perform some specific operation like searching or sorting.)
7. **Merging**

* **Abstract Data Types:**

It tells us what operations are to be performed but now how these operations will be implemented.

It does not specify how data will be organized in memory and what algorithms will be used for implementing the operations.

It is called abstract as it gives an implementation-independent view.

The process of providing only the essentials and hiding details is known as abstraction.

* **Advantages of Data Structures:**

1. Efficiency
2. Reusability
3. Abstraction

* **DS 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.

**Characteristics of an Algorithm:**

1. Input
2. Output
3. Unambiguity: Instructions should be clear and simple
4. Finiteness: Limited number of instructions
5. Effectiveness
6. Language independent

**Dataflow of an Algorithm**

1. Problem
2. Algorithm
3. Input
4. Processing Unit
5. Output

**Factors**

1. Modularity: Can be broken down into small modules or small steps
2. Correctness: Inputs produce desired output
3. Maintainability
4. Functionality: It considers various logical steps to solve real-world problem
5. Robustness
6. User-friendly
7. Simplicity
8. Extensibility

**Approaches of Algorithm**

1. **Brute force algorithm:** It is also known as exhaustive search algorithm, It searches all possibilities to provide the required solution. It has two types
2. Optimizing: Finding all the best solutions of a problem and then take out the best solution.
3. Sacrificing: As soon as the best solution is found, then it will stop.
4. **Divide and Conquer:** It is a step-by-step design.

It breaks down the algorithm to solve the problem in different methods. It allows you to break down the problem into different methods.

Valid output is produced for valid input and this output is passed to some other function

1. **Greedy Algorithm:** It is an algorithm paradigm that makes an optimal choice on each iteration with the hope of getting the best solution. It is easy to implement and has faster execution time.
2. **Dynamic Programming:**
3. It breaks down the problem into a subproblem to find the optimal solution.
4. After breaking down the problem, it finds the optimal solution out of these subproblems.
5. Stores the result of the subproblems is known as memorization.
6. Reuse the result so that it cannot be recomputed for the same subproblems.
7. Finally, it computes the result of the complex program.

**5) Branch and Bound Algorithm:** It can only be applied to integer

programming problems. This approach divides all the sets of feasible

solutions into smaller subsets. These subsets are further evaluated to

find the best solution.

1. **Randomized Algorithm:** When a random variable is introduced in an algorithm. Some random bits are introduced by the algorithm and added in the input to produce the output, which is random in nature.
2. **Backtracking:** It is an algorithmic technique that solves the problem recursively and removes the solution if it does not satisfy the constraints of a problem

**Algorithm Analysis**

1. **Priori Analysis :** It is done before implementing the algorithm.
2. **Posterior Analysis:** It is a practical analysis of an algorithm.

**Algorithm Complexity:**

1. **Time Complexity:** Amount of time required to complete the execution. It is denoted by O. It is calculated by counting the number of steps to finish the execution
2. **Space Complexity:** Amount of space required to solve a problem and produce an output. It is also represented by O. For an algorithm, space is required to store instructions, constant values, variable values and to track function calls, jumping statements etc.

Auxiliary space: The extra space required by the algorithm, excluding the input size is known as an auxiliary space.

Space complexity = Auxiliary space + Input Size

**Types of Algorithm**

1. **Search Algorithm:**
2. **Linear Search:** Very simple algorithm that starts searching for an element or value from the beginning of an array until the required element is not found. It compares the element to be searched with all the elements in an array. If the match is found, it returns the index of the element else it returns -1. It can be implemented in an unsorted list.
3. **Binary Search:** It is used to search the element from the sorted list. Elements must be stored in sequential order or the sorted manner to implement binary algorithm. It is implemented in a sorted list. It finds the middle element of the list.
4. **Sorting Algorithm:**

Sorting algorithms are used to rearrange the 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.