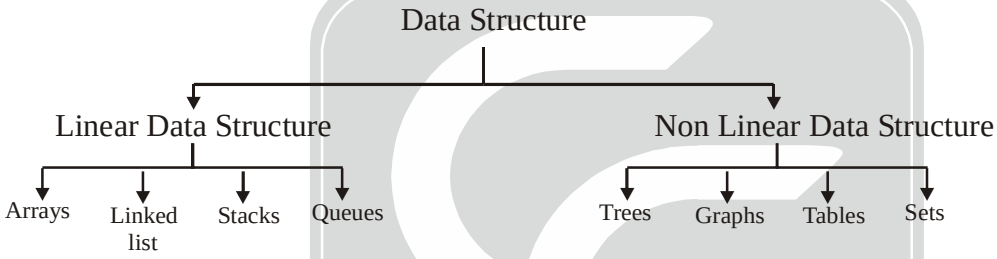
Chapter 1

Introduction to Data Structure & Algorithms

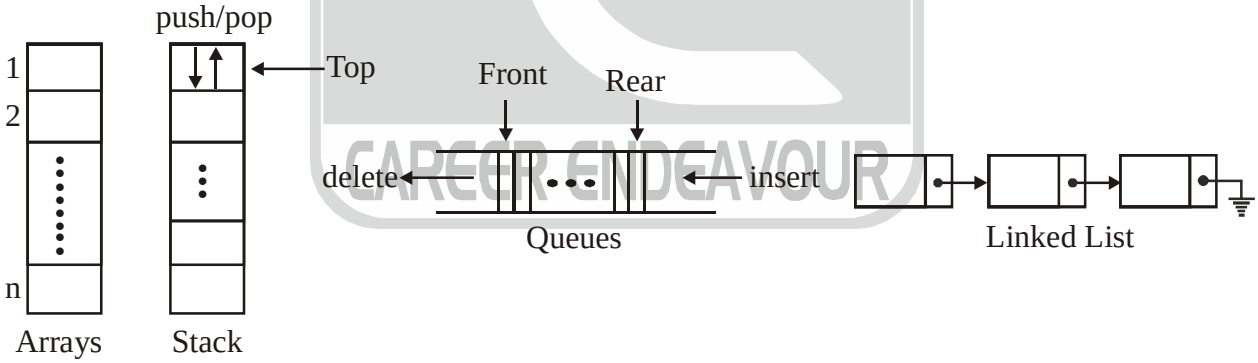
Data Structure :

A data structure is an organization of the data to solve a problem in such a way that data can be accessed efficiently by a progam. The choice of particular data structure depends upon the following consideration.

- It must be able to represent the inherent relationship of data in the real world.
- It must be simple enough so that it can be proceesed efficiently when necessary.

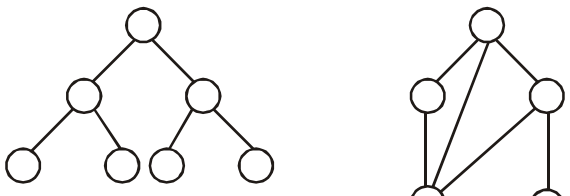


Linear Data Structure : A Data structure in which there is a linear order among the data elements. The elements form a sequence. For example: Arrays, Stacks, queues, linked list.



Non-Linear DataStructure: The elements doesn't form a sequence. A Data structure in which there is no linear order among the data elements.

Example: Tree, Graph.



Tree

Graph

Algorithm: The essence of a computational procedure in step by step instruction.

Program: An implementation of an algorithm in some programming langugage.

Properties:

Every algorithm must satisfy the following criteria.

- **Input:** There are zero or more values which are externally supplied.
- **Output:** At least one value is produced.
- **Definition:** Each step must be clear and unambigious. For same input always get same output.
- **Finiteness:** Algorithm must terminate after a finite number of steps.
- **Effectiveness:** Each step must be sufficiently basic so that it can be carried out by a person using only paper and pencil.
- **Independence :** It should be independent of any programming language.

Steps to construct an algorithm:

Steps to construct an algorithm
(a) Problem defintion : Understand the problem clearly.
(b) Designing the algorithm: Use various designing methods e.g. greedy, dynamic programming etc.

(c) Flow chart : Construct a flow chart

(d) Verification : Check the outputs for different inputs

(e) Implementation : Convert the algorithm to a compatible language

(f) Analysis : Analyze the algorithm for time and space.

Algorithm describes actions on the input instances. Infinitely many correct algorithms for the same algorithmic problem are possible. Any algorithm which is efficient in terms of the following is good for us.

(a) **Running time:** Good algorithms takes least time.
(b) **Space used:** Good algorithm occupies least space.

Analysis :

Analysis is done on two basis :
(i) Time (ii) Space
Note : Always first priority is given to the time.
Time Complexity : We have an algorithm A, then time complexity.

$$T(A) = C(A) + R(A)$$

Where, T (A) is time complexity, C(A) is compile time and R(A) is run time.
Algorithm: It is step by step process to solve a problem.