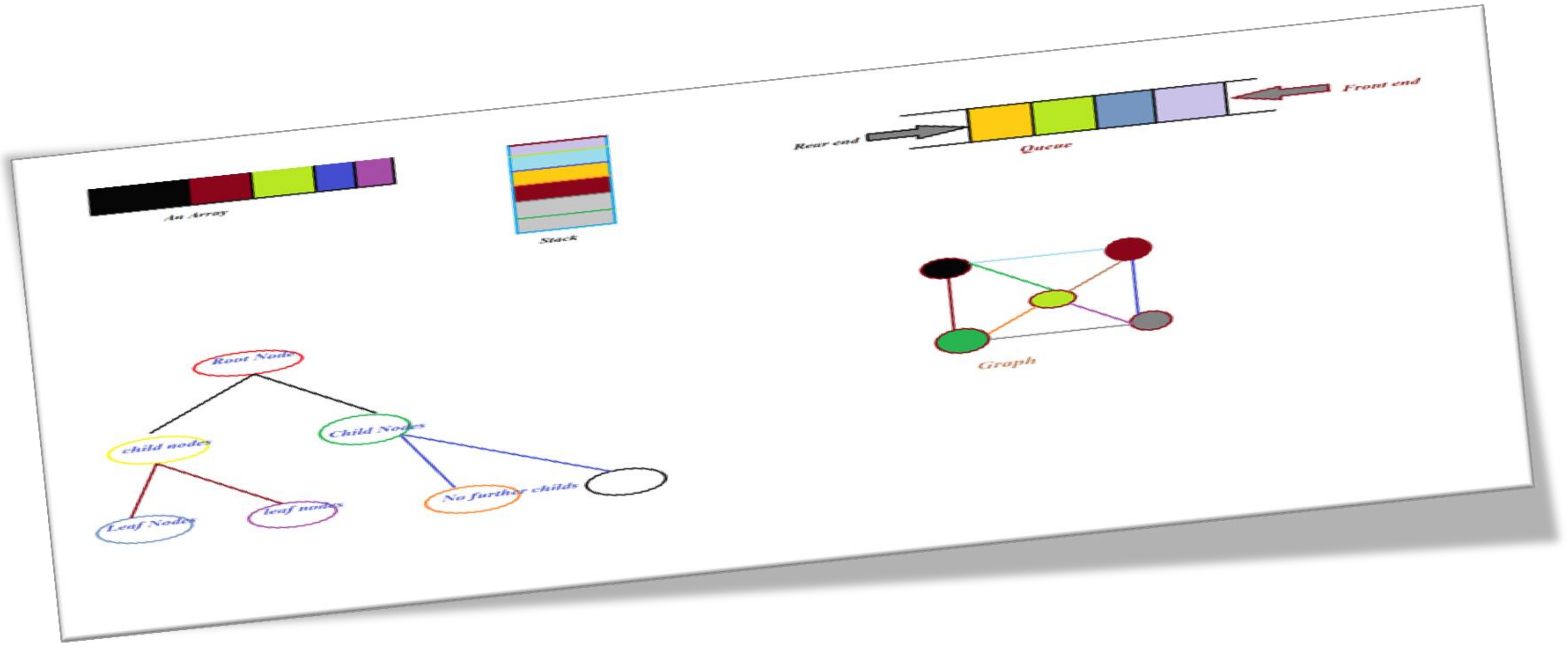
Data Structure



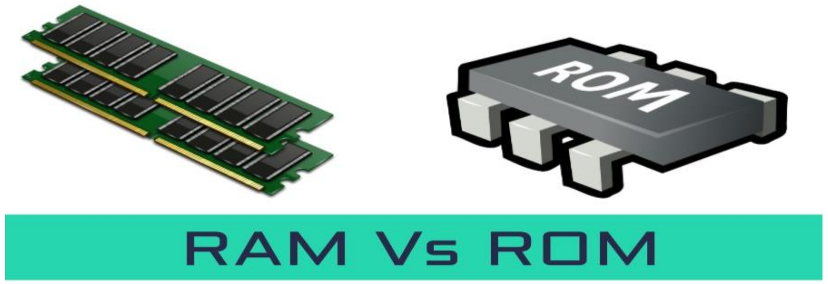
PART 1

الجامعة المستنصرية / كلية العلوم / قسم علوم الحاسوب المرحلة الثانية / مسائي

Introduction to Data Structure

- **Computer is a electronic machine which used for data processing and manipulation.**
- **When programmer collects such type of data for processing, he would require to store all of them in computer's memory.**
- **In order to make computer work we need to know:**
  1.Representation of data in computer.
  2.Accessing of data.
  3.How to solve problem step by step.

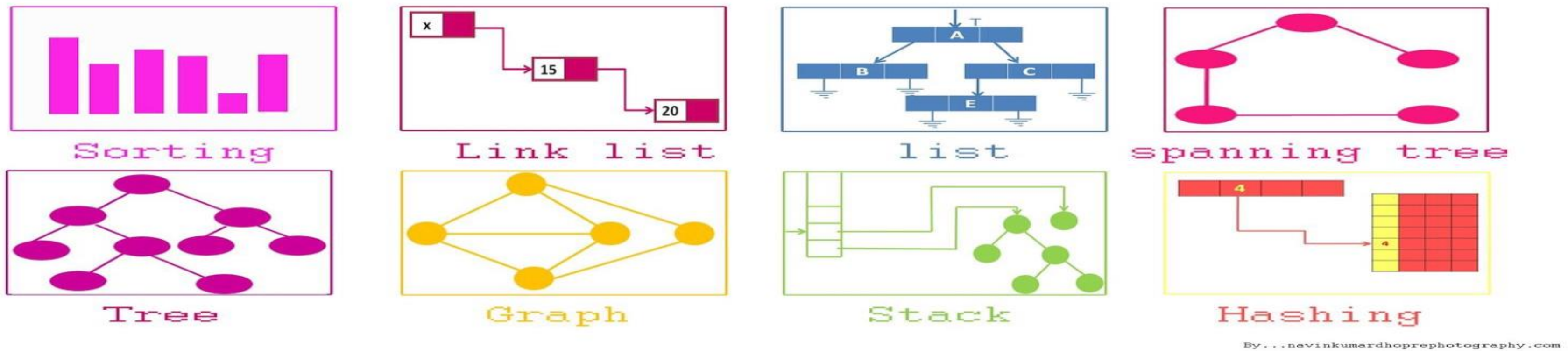
- **For doing this task we used data structure.**



What is Data Structure

- **Data structure is a representation of the logical relationship between individual elements of data.**

- **Data structure is a way of organizing all data items that considers not only the elements stored but also their relationship to each other.**
- **We can also define data structure as a mathematical or logical model of a particular organization of data items.**
- **The representation of particular data structure in the main memory of a computer is called as storage structure.**
- ***It is defined as the way of storing and manipulating data in organized from so that it can be used efficiently.***



- **Data structure mainly specifies the following four things:**
  1. **Organization of data.**
  2. **Accessing methods.**
  3. **Degree of associativity.**
  4. **Processing alternatives for information.**

- **Data structure study covert the following points:**
  1. **Amount of memory require to store.**
  2. **Amount of time require to process.**
  3. **Representation of data in memory.**
  4. **Operations performed on that data.**

Algorithm:- Is a finite set of instructions which, if followed, accomplish a particular task. In our course, it is not enough to specify the form of data structure, but we must give also the algorithm of operation to access these data structures. Any algorithm must satisfy the following criteria:

1- Input.   2- Output.   3- Definiteness.

4- Finiteness.   5- Effectiveness.

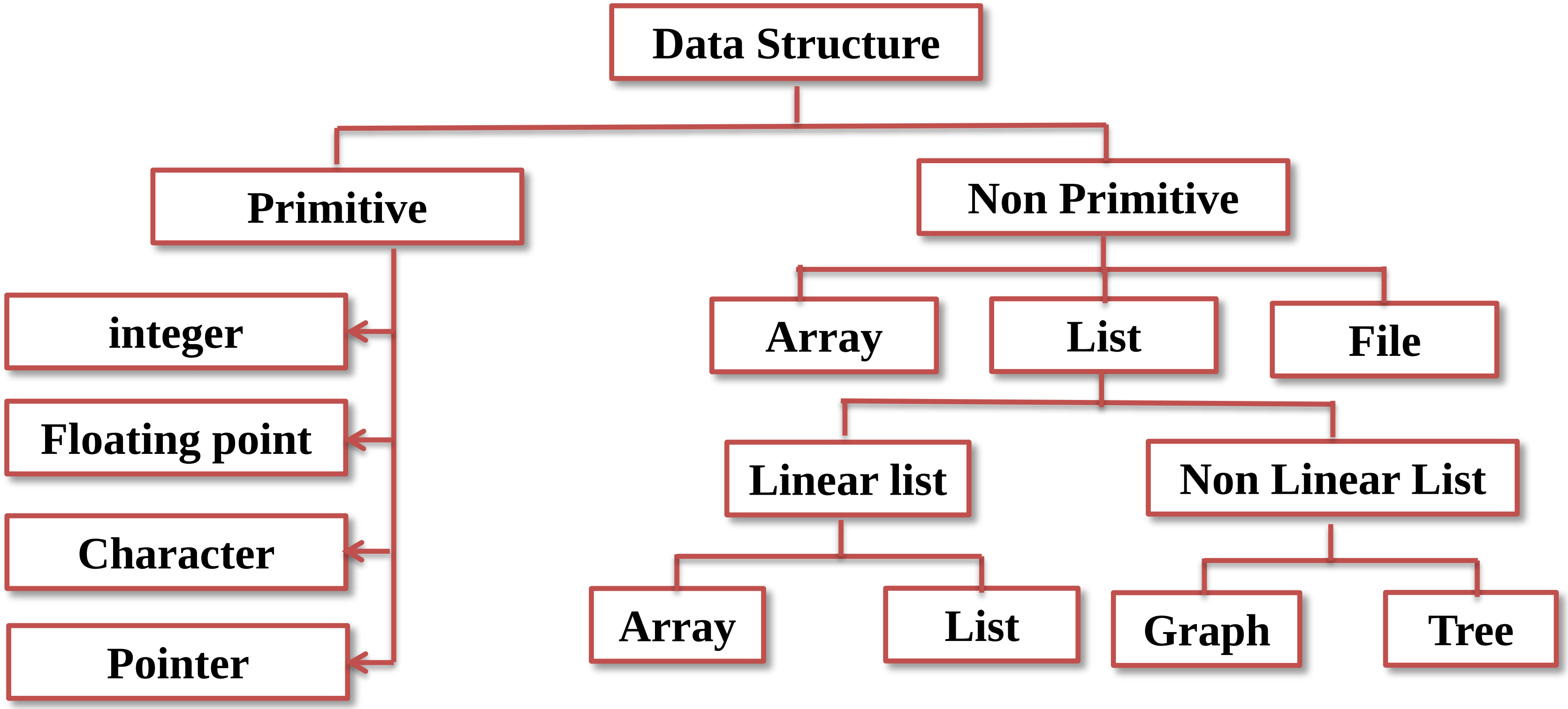
Algorithm + Data Structure = Program

How to Choose the Suitable Data Structure

Data structure is a particular way of organizing data in a computer. The developer must choose the appropriate data structure for better performance. If the developer chooses bad data structure, the system does not perform well. ***To choose the suitable data structure, we must use the following criteria:***

1- Data size and the required memory.
2- The dynamic nature of the data.
3- The required time to obtain any data element from the data structure.
4- The programming approach and the algorithm that will be used to manipulate these.

Classification of data Structure



Data structures are normally classified into two broad categories :

1. **Primitive data structure.**
2. **Non primitive data structure.**

Data Type

A particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

Primitive Data Structure

- **Primitive data structure are basic structures and are directly operated upon by machine interactions.**
- **Primitive data structures have different representations on different computers.**
- **Integers, floats, character and pointers are examples of it.**
- **These data types are available in most programming languages as built in type.**

Integer: **it is a data type which allows all values without fraction part. We can use it for whole numbers.**

Float: **it is a data type which use for storing fractional numbers.**

Character: **it is a data type which is used for character values.**

Pointer: **a variable that holds memory address of another variable are called pointer.**

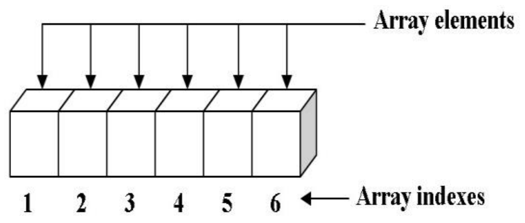
Non Primitive Data Structure

- These are more sophistical data types.
- These are derived from primitive data structures.
- The non-primitive data structures emphasize on structuring of a group of homogeneous or heterogeneous data items.
- Examples of non-primitive data type are Array, List, and File etc.
- A non primitive data type is further divided into Linear and Non-Linear data structure.

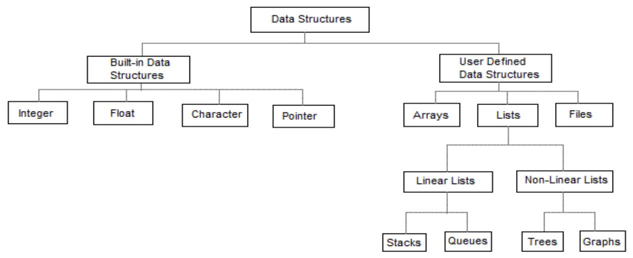
Array : **An array is a fixed size sequenced collection of elements of the same data type.**

List : **An ordered set containing variable number of elements is called as lists.**

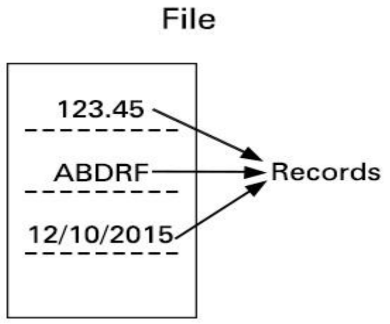
File : **A file is a collection of logically related information. It can be viewed as a large list of records consisting of various fields.**



One-dimensional array with six elements



INTRODUCTION TO DATA STRUCTURES

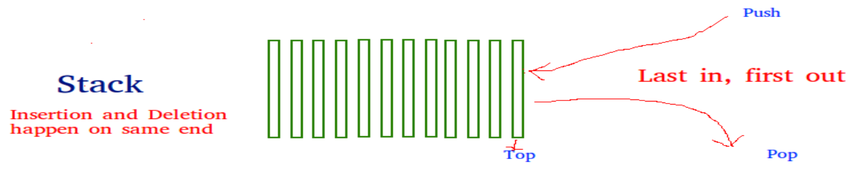


Linear Data Structure

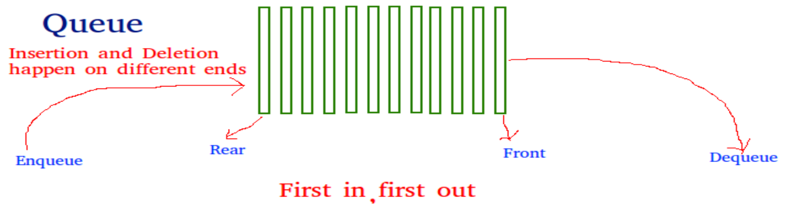
- A data structure is said to be ***Linear***, if its elements are connected in linear fashion by means of logically or in sequence memory locations.
- There are two ways to represent a linear data structure in memory:
  1. static memory allocation.
  2. Dynamic memory allocation.
- The possible operations on the linear data structure are: ***Traversal***, ***Insertion***, ***Deletion***, ***Searching***, ***Sorting*** and ***Merging***.

- Example of linear data structure are ***Stack*** and ***Queue***.
- A non primitive data type is further divided into **Linear** and **Non-Linear** data structure.

Stack : stack is a data structure in which insertion and deletion operations are performed at on end only.



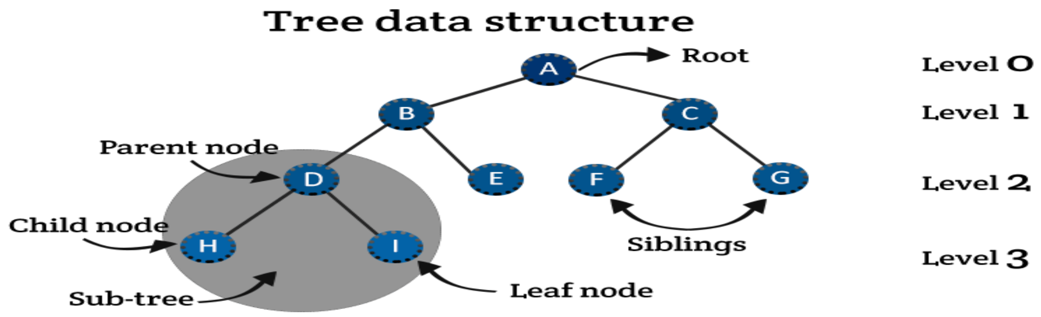
Queue : the data structure which permits the insertion at one end and deletion at another end.



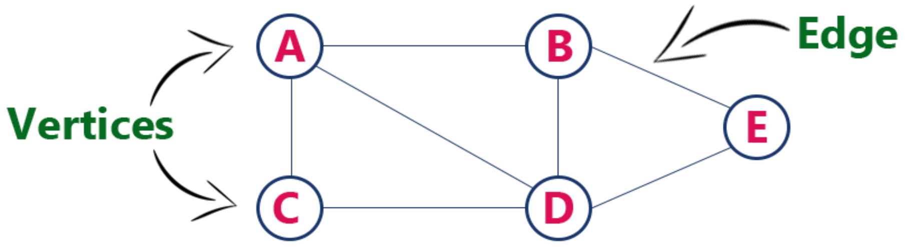
NonLinear Data Structure

- Nonlinear data structure are those data structure in which data items are not arranger in sequence.
- Examples of nonlinear data structure are *Tree* and *Graph*.

Tree : A tree can be defined as finite set of data items (nodes) in which data items are arranged in branches and sub branches according to requirement.



Graph : Graph is a collection of nodes (information) and connecting edges (logical relation) between nodes.



Difference between linear and Non Linear Data Structure

Linear Data Structure	Non Linear Data Structure
Every item is related to its pervious and next time.	**Every item is attached with many other items.**
Data is arranged in linear sequence.	**Data is not arranged in sequence.**
Data items can be traversed in a single run.	**Data cannot be traversed in a single run.**
Eg. Array, Stacks, Linked list, Queue	**Eg. Tree, Graph.**
Implementation is easy.	**Implementation is difficult.**