Lecture 1- Introduction to Data Structures

Data Structure: is the link between the programmer's view of data and the implementation of these data in computers. It consists of a storage method and one or more algorithm that used to access or modifies data.

Data structure = Data+ Algorithm

Data structure can be also defined as a particular way of organizing data in a computer so that it can be retrieved and used efficiently.

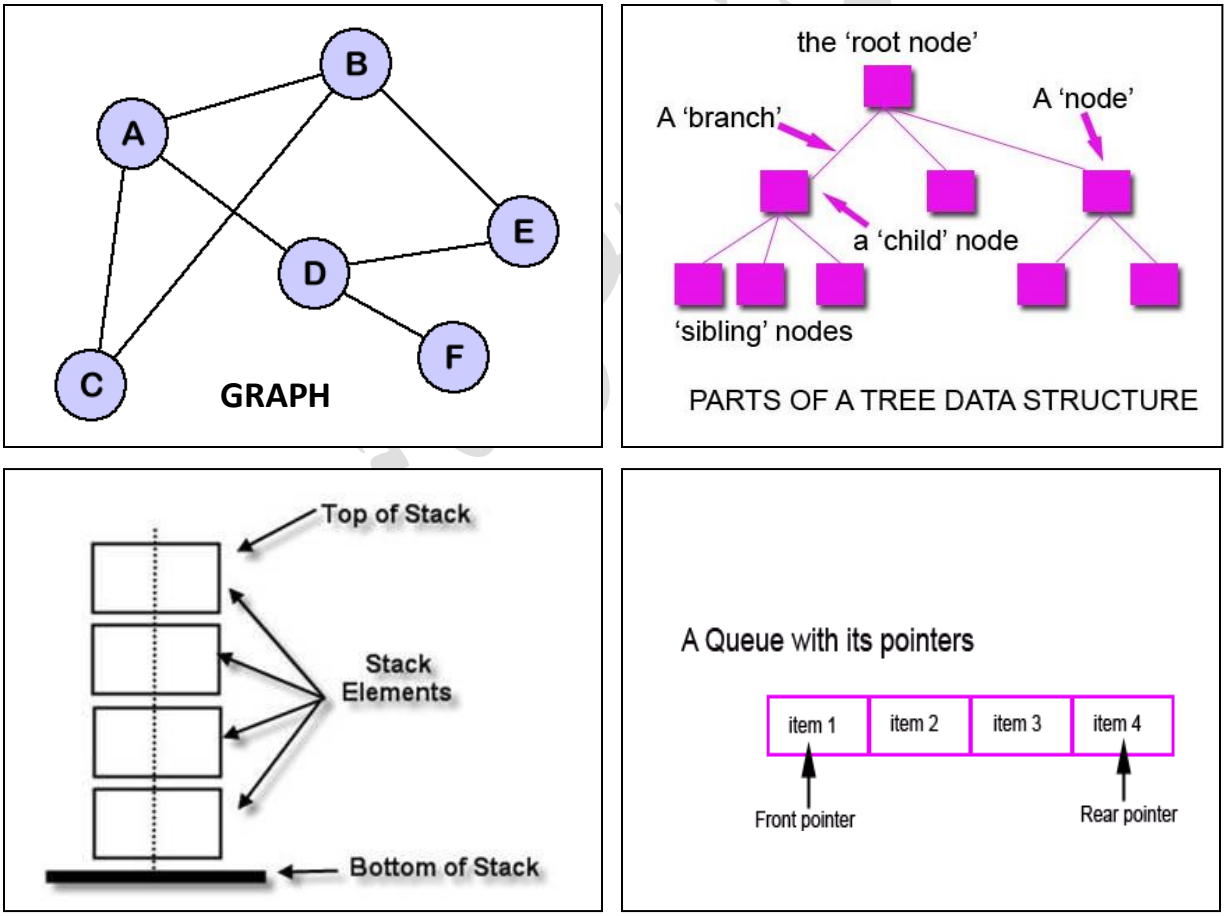

Figure 1. Different types of data structures

Criteria to choose correct Data Structure:

a) The size of data will be used.

b) The way that data will be used.

c) How often the data will be changed.

d) The time required to access any information in the data structure.

e) Required capacity.

f) The programming language/ method will be used.

Primitive and Non-Primitive data Types

Data type specifies the type of data stored in a variable. The data type can be classified into two types, primitive data type and non-primitive data type.

Primitive Data types: are the basic data types that are available in most of the programming languages. The primitive data types are used to represent single values like:

Integer: This is used to represent a number without decimal point.

Example: 12, 90

Float and Double: This is used to represent a number with decimal point.

Example: 45.1, 67.3

Character: This is used to represent single character

Example: ‘C’, ‘a’

String: This is used to represent group of characters.

Example: "CIHAN University"

Boolean: This is used represent logical values either true or false.

Non-Primitive Data types: are derived from primary data types and used to store group of values like:

- Arrays

- Structure
- linked list
- Stacks
- Queue

Operations that can be performed on data structures:

1. Traversing: It is used to access each data item exactly once so that it can be processed.

2. Searching: It is used to find out the location of the data item if it exists in the given collection of data items.

3. Inserting: It is used to add a new data item in the given collection of data items.

4. Deleting: It is used to delete an existing data item from the given collection of data items.

5. Sorting: It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data.

6. Merging: It is used to combine the data items of two sorted files into single file in the sorted form.

Types of data structures

There are two types of data structures, these are:

Linear Data Structure: Is the data structure that every element is linked with the next one sequentially. Since the data items are arranged in sequence. Samples of a linear data structure are the stack and the queue.

Nonlinear data structure: is the data structure where the element may attach to more than one element and data items are not in sequence. A sample of nonlinear data structure is a tree.

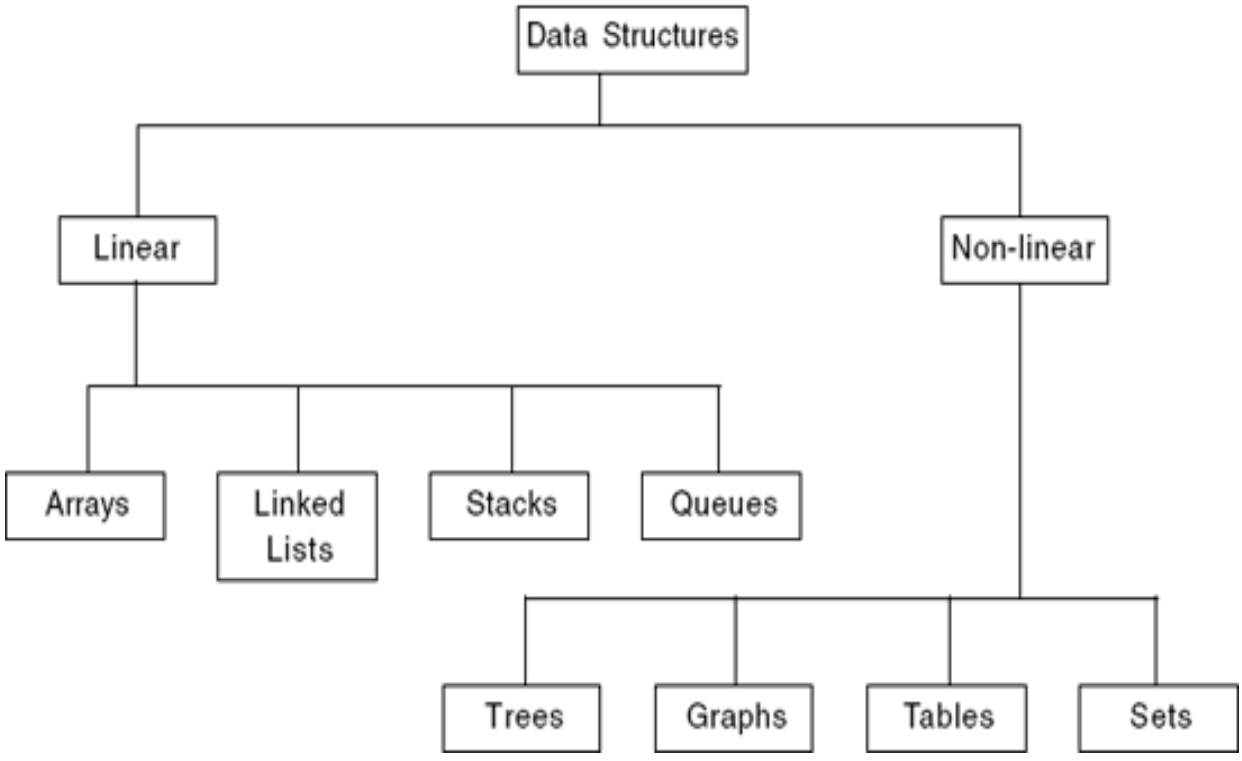

Figure 2. Types of data structures

Storage Allocation:

It's the way in which the data items are physically stored in the memory, so there are the following two types

1-sequential allocation: it's the simpler way in which the data items are stored in continuous memory allocations one by one. Arrays are the data types that are used to implement such types of storages.

Sequential Allocation Advantages:

- Easy to implement.
- Less storage space.

Sequential Allocation Disadvantages:

- It needs a sequential free space in memory.
- It required that we know previously the max number of items that will be used.
- Over flow problem is occasionally occurs.

2-Dynamic Allocation: implementation of data structure is the dynamic linking; since there is no sequential allocation but every data item has the address of the next one instead.

Here each data item will be called as "node"; this node will have two parts, one carried the required information and the other is for the next item address.

Dynamic Allocation Advantages:

- Easy to add or remove.
- No over flow problem.
- We do not need to specify the number of nodes previously.

Dynamic Allocation Disadvantages:

- More storage space for each data item, since each one must has a value and the information of the next address as well.
- Complicated and hard to achieve random access.