هياكل البيانات

المرحلة الثانية

محاضرة (1)

م.م فرح معاذ جاسم

Farahmaath86@uoanbar.edu.iq

1.1 DATA STRUCTURES

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.

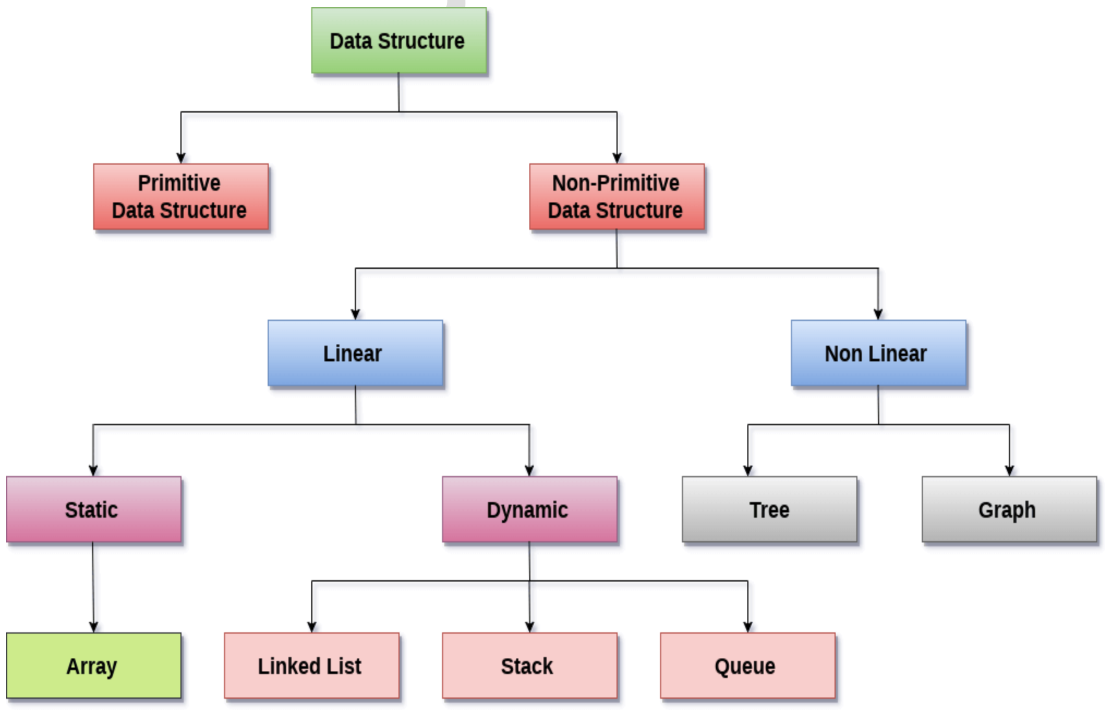

Figure (1) represented the data structures classification

1.2 PRIMITIVE DATA STRUCTURES

Data type is a way to classify various types of data such as integer, string, etc. which determines the values that can be used with the corresponding type of data, the type of operations that can be performed on the corresponding type of data.

Built-in Data Type

Those data types for which a language has built-in support are known as Built-in Data types. For example, most of the languages provide the following built-in data types.

- Integers
- Boolean (true, false)
- Floating (Decimal numbers)
- Character and Strings

1.3 NON-PRIMITIVE DATA STRUCTURES (DERIVED DATA TYPE)

Those data types which are implementation independent as they can be implemented in one or the other way are known as derived data types. These data types are normally built by the combination of primary or built-in data types and associated operations on them. For example −

- List
- Array
- Stack
- Queue

1. **Linear Data Structures:** A data structure is called linear if all of its elements are arranged in the linear order. In linear data structures, the elements are stored in non-hierarchical way where each element has the successors and predecessors except the first and last element.

Types of Linear Data Structures are given below:

A. **Arrays:** An array is a collection of similar type of data items and each data item is called an element of the array.

B. **Stack:** Stack is a linear list in which insertion and deletions are allowed only at one end, called **top**.

C. **Queue:** Queue is a linear list in which elements can be inserted only at one end called **rear** and deleted only at the other end called **front**.

D. **Linked List:** Linked list is a linear data structure which is used to maintain a list in the memory. It can be seen as the collection of nodes stored at non-contiguous memory locations. Each node of the list contains a pointer to its adjacent node.

2. **Non Linear Data Structures**

This data structure does not form a sequence i.e. each item or element is connected with two or more other items in a non-linear arrangement. The data elements are not arranged in sequential structure. Types of Non Linear Data Structures are given below:

A. **Trees:** Trees are multilevel data structures with a hierarchical relationship among its elements known as nodes. The bottommost nodes in the hierarchy are called **leaf node** while the topmost node is called **root node**. Each node contains pointers to point adjacent nodes.

B. **Graphs:** Graphs can be defined as the pictorial representation of the set of elements (represented by vertices) connected by the links known as edges. A graph is different from tree in the sense that a graph can have cycle while the tree can not have the one.

1.4 OPERATIONS ON DATA STRUCTURE

1. Traversing
2. Insertion
3. Deletion
4. Searching
5. Sorting
6. Merging

1.5 HOW TO CHOOSE THE SUITABLE DATA STRUCTURE

For each set of, data there are different methods to organize these data in a particular data structure. 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.