

Data Structure

Computer science has a subfield called as data structures. Understanding data organization and flow management by studying data structures enables us to optimize any process or program's efficiency. Data Structure is a specific method of storing and organizing data in the computer's memory to retrieve and effectively use it later quickly. A data structure is a logical or mathematical representation of a certain type of data organization. Data can be managed in a variety of ways.

A specific data model's scope is determined by two things:

- The data must first be sufficiently put into the structure to indicate a clear association between the data and a physical item.
- Second, the structure should be so simple that anyone may learn how to analyze the data effectively as needed.

Data structures include things like arrays, linked lists, stacks, queues, trees, etc. Nearly every area of computer science, including compiler design, operating systems, graphics, artificial intelligence, and many others, makes extensive use of data structures.

The essential component of many algorithms in computer science is a data structure, which enables programmers to manage data efficiently. As the primary goal of the software is to save and retrieve user data as quickly as possible, it is essential to enhance the performance of an application or piece of software.

When discussing data structures, the following basic terms are frequently used:

1. Data: A collection of values or an elementary value are both acceptable definitions of data. Examples of employee-related data include the name and ID of the employee.
2. Data Items: Data Items are single units of value.

3. Group Items: Group Items are data items that have subordinate data items. An employee's name, for instance, could include their first, middle, and last names.
4. Elementary Items: Data items classified as elementary cannot be subdivided into other items. Using an employee's ID as a demonstration.
5. Entity and Attribute: An entity represents a class of specific objects. It is made up of several Attributes. Each attribute represents a particular attribute of that entity.

Classification of DATA STRUCTURES

Data structures are generally classified into

Primitive data Structures

Non-primitive data Structures

1. Primitive data Structures: Primitive data structures are the fundamental data types which are supported by a programming language. Basic data types such as integer, real, character and Boolean are known as Primitive data Structures. These data types consists of characters that cannot be divided and hence they also called simple data types.

2. Non- Primitive data Structures: Non-primitive data structures are those data structures which are created using primitive data structures. Examples of non-primitive data structures is the processing of complex numbers, linked lists, stacks, trees, and graphs.

Based on the structure and arrangement of data, non-primitive data structures is further classified into

1. Linear Data Structure
2. Non-linear Data Structure

1. Linear Data Structure:

A data structure is said to be linear if its elements form a sequence or a linear list. There are basically two ways of representing such linear structure in memory.

1. One way is to have the linear relationships between the elements represented by means of sequential memory location. These linear structures are called arrays.
2. The other way is to have the linear relationship between the elements represented by means of pointers or links. These linear structures are called linked lists.

The common examples of linear data structure are Arrays, Queues, Stacks, Linked lists

2. Non-linear Data Structure:

A data structure is said to be non-linear if the data are not arranged in sequence or a linear. The insertion and deletion of data is not possible in linear fashion. This structure is mainly used to represent data containing a hierarchical relationship between elements. Trees and graphs are the examples of non-linear data structure.

Basic Operations on Data Structures

The data appearing in data structures are processed by means of certain operations.

The following four operations play a major role in this text:

1. Traversing: accessing each record/node exactly once so that certain items in the record may be processed. (This accessing and processing is sometimes called “visiting” the record.)
2. Searching: Finding the location of the desired node with a given key value, or finding the locations of all such nodes which satisfy one or more conditions.

3. Inserting: Adding a new node/record to the structure.

4. Deleting: Removing a node/record from the structure.

The following two operations, which are used in special situations:

1. Sorting: Arranging the records in some logical order (e.g., alphabetically according to some NAME key, or in numerical order according to some NUMBER key, such as social security number or account number)

2. Merging: Combining the records in two different sorted files into a single sorted file.