DSA
INTERNSHALA / NESO Academy
INTRODUCTION TO DATA STRUCTURE

What is Data?

The quantities, characters or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical or mechanical recording media.

When Data becomes Information?

If data is arranged in a systematic way then it gets a structure and become meaningful.

This meaningful or processed data is called Information.

Its not difficult to understand that the data needs to be managed in such a way so that it can produce some meaningful information.

To provide an appropriate way to structure the data, we need to know about Data Structures.

What is Data Type?

Two important things about data types:

1. Defines a certain domain of values.
2. Defines operations allowed on those values.

User Defined Data Types

In contrast to primitive data types, there is a concept of used defined data types.

The operations and values of user defined data types are not specified in the language itself but is specified by the user.

Examples: Structure, union and enumeration.

Abstract Data Types

Abstract Data Types are like user defined types which defines a operations on values using functions without specifying what is there inside the function and how the operations are performed.

Examples: Stack – initialize(), Push(), Pop(), isEmpty(), isFull().

Think of ADT as a black box which hides the inner structure and design of the data types from the user.

There are multiple ways to implement an ADT.

Why ADT?

The program which uses data structure is called a client program.

It has access to the ADT, i.e., interface.

The program which implements the data structure is known as the implementation.

Advantage:

Let say, if someone wants to use the stack in the program, then he can simply use push and pop operations without knowing its implementation.

Also, if in future, the implementation of stack is changed from array to linked list, the client program will work in the same way without being affected.

ADT provides Abstraction.

What is Data Structure?

A data structure is the systematic way to organize data so that it can be used efficiently.

Example:

Arrays – Instead of creating multiple variables of same type, why not create an array to store all the values.

Storing strings is equivalent to storing a sequence of characters. This requires an array.

ADT tells us what is to be done and data structure tells us how to do it.

In reality, different implementations of ADT are compared for time and space efficiency. The one best suited according to the current requirement of the user will be selected.

Real Life Examples:

Stack data structure is used in implementing redo and undo feature.

Arrays are used to store a bitmap image. Bitmap images are stored as a series of tiny dots called pixels. Here, each pixel is actually a small square assigned a colour and then arranged in a pattern to form an image.

Graphs are used in storing friends and connections on social sites.

Advantages of Data Structure:

- Efficiency – proper choice of data structure make program efficient in terms of space and time.
- Reusability – one implementation can be used by multiple client programs.
- Abstraction – Data structure is specified by an ADT which provides a level of abstraction. The client program doesn’t have to worry about the implementation details.

Static Data Structures

In these type of data structures, the memory is allocated at compile time. Therefore, maximum size is fixed.

Advantage – Fast access.

Disadvantage – Slower insertion and deletion.

Dynamic Data Structures

In these type of data structures, the memory is allocated at run time. Therefore, maximum size is flexible.

Advantage – Faster insertion and deletion.

Disadvantage – Slower access.

Linear Data Structures

- Array:
  - An array is a finite set of homogeneous elements stored in contiguous memory locations.
  - An array is highly useful to store multiple values belonging to same data type.
  - An array of n elements is indexed form 0 to n-1.
  - Starting subscript might vary from language to language.
- Matrices (2D array):
  - A 2D array is a collection of rows.
  - A matrix of size n*m will have indexes as [0…n-1] [0…m-1]

    An array name is a pointer to the base address. Hence (arr) or (arr+0) represents the address of the very first element. As the first element is the 0th location away from the base address, hence subscript starts from 0. And unsigned int data type range also starts from 0.

- Stacks:
  - A stack is a data structure that follows the principle of *LIFO* (Last in First Out).
  - Inserting an element into a stack is referred to as *pushing*.
  - Deleting an element from a stack is referred to as *popping*.
  - A new element is pushed only on the top.
  - An element can be deleted only form the top.
- Queue:
  - A queue is a data structure that follows the principle of FIFO (first in first out).
  - There are two ends in a queue – front and rear.
  - An element is always removed from the front.
  - A new element is always inserted from the rear.
- Linked Lists:
  - Three types:
    - Linear linked list
    - Doubly linked list
    - Circular linked list

Non-Linear Data Structures

- Trees:
    - Binary Tree
        - Every parent node can have at maximum of two child nodes.
- Graphs:
    - Directed graphs
    - Undirected graphs

Brief Intro to Computer Memory

- Computer memory is similar to human brain. It is used to store data and computer programs.
- The computer memory is stored in extremely large number of cells.
- Each memory cell has a unique address.
- Types:
    - Primary memory
        - Main memory
        - Access at lightening speed
        - Quite costly
        - Much less than secondary memory
    - Secondary memory
        - Mass storage media
        - Stores voluminous data
        - Much slower
        - Much cheaper
- Types of Primary memory:
    - RAM (Random Access Memory)
        - Temporary memory
        - Data lost if power supply is off
        - Data can be processed only in RAM
        - Variables, arrays, stacks, queues, linked list, trees & graphs etc. are in RAM only.
    - ROM (Read Only Memory)
        - Permanent memory
        - Data not lost if power supply is off
        - Manufacturer writes info & hardwired programs
        - Cannot be overwritten by users in normal circumstances
- Types of Secondary Memory:
    - HDD (Hard Disks Drives)
        - The most commonly used for mass storage.
        - Magnetic recording of data by magnetising a thin film on the surface.
    - SSD (Solid State Drive)

        - Persistent flash memory.
        - Much faster than HDD.
        - Earlier used for mobile phones but now popular with computers also.
    - USB Drives (Universal Serial Bus)
        - Extremely popular.
        - Provides huge data storage despite being physically small.
        - Removable and rewritable.
    - Optical Drives
        - Reading and writing using lasers.
        - CD’s, DVD’s, Blue rays are examples.
    - Magnetic tapes
        - Serial access storage devices and much slower.
        - Voluminous data can be stored.
        - Mostly useful for taking backups of data.