Data Structures & Algorithms

Introduction

What is data?

A collection of facts from which conclusion may be drawn. e.g. Data: Temperature 35°C;

Conclusion: It is hot.

Types of data

Textual: For example, your name (Muhammad)

Numeric: For example, your ID (090254)

Audio: For example, your voice

Video: For example, your voice and picture

Motivation

Why data structures?

A famous equation

Programs = Data Structures + Algorithms

Data Structures are used to write efficient programs.

Data structure: method to store and retrieve data efficiently.

Algorithm: method for solving a problem.

Learning Data Structures and Algorithms allows us to write efficient and optimized computer programs.

What is Data Structure?

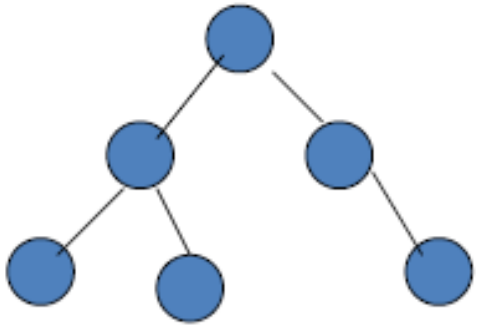
A data structure is a particular way of storing and organizing data in computer to facilitate faster access of data.

Examples of more commonly used data structuring mechanisms includes array, stack, queue, linked list, tree, graph, hash table, etc.

Rarely is one data structure better than another in all situations, i.e. no single data structure works well for all purposes.

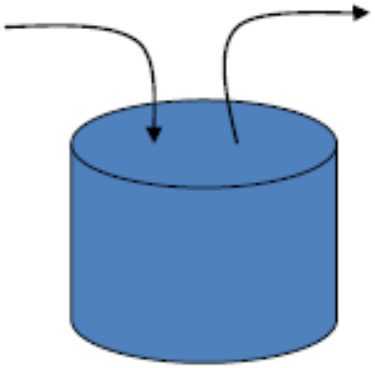
Array

Linked List



Tree

Queue



Stack

Basic Data Structures

Classification of Data Structures

Data Structures can further be classified into two categories:

Linear Data Structure

(Array, Linked List, Stack, Queue, etc)

Non-Linear Data Structure

(Tree, Graph, Hash table, etc)

Classification of Data Structures

In linear data structures every data elements form a sequence, and every data elements has a unique predecessor and successor, except the first and the last elements, whereas in case of non-linear data structures elements does

not form a sequence, there is no such unique predecessor and unique successor restriction exists, elements may be arranged in any desired fashion.

Static and Dynamic Data Structures

A data structure may be static or dynamic.

A static data structure has a fixed/finite size/capacity. It is created during compilation time; once we define the number of elements it can hold, the number doesn't change but can change content of a static data structure.
Example – Array is static

A dynamic data structure grows and shrinks at run time as required by its contents. A dynamic data structure is implemented using links/ pointers.
Example - Linked list, Stack, Queue, Tree, Graph, Perfect Hash Table

Basic operations in data structure

Traversal – access/process each data item exactly once

Insertion - add a new data item in the given list of data items

Deletion - delete an existing data item from the given list of data items

Merging - combine the data items of two sets of sorted list of data items into single list of data items in the sorted form

Splitting – divide the data items of a single list of data items into two sets of list of data items

Concatenation - Given two sets of list of data items, appends one list of data items after another list of data items