Fundamental Data Structures

Madiba Hudson-Quansah

Contents

Chapter 1

Introduction to Data Structures

1.1 Data Structures

> **Definition 1.1.1: Data Type**
>
> Describes the set of values that a variable can hold and the types of operations that can be performed on it.

> **Definition 1.1.2: Data Structure**
>
> A way of storing and organizing data in computer memory / The logical mathematical model of a particular organization of data.

In choosing a data structure, we consider the following in the following order:

1. The relationships of the data in the real world
2. The type and amount of data to be stored
3. Cost (time complexity) of operations
4. Memory occupations
5. Ease of implementation

1.1.1 Types of Data Structures

Data structures can be broadly classified into two types:

- Primitive Data Structures
- Non-Primitive / Composite Data Structures

With data structures under the primitive category being:

- Variables
  - Integer
  - Float
  - Character
  - Boolean
  - Enumerated
  - Reference / Pointer

And data structures under the non-primitive category being:

- Arrays

- Structure / Record

- Union

- Class

- Abstract Data Type

  - Linear

    * List

    * Stack

    * Queue

  - Non-Linear / Associative

    * Tree

    * Graph

    * Hash Table

Data structures can also more simply be classified in two ways:

- Linear / Sequential Data Structures

  - Linked List

  - Array

  - Stack

  - Queue

  - Set

- Non-Linear / Associative Data Structures

  - Graph

  - Tree

  - Hash Table

In this way data structures are defined by an implementation of a particular abstract data type.

1.1.1.1 Primitive Data Structures

These are the basic data structures that are directly operated on by machine / CPU instructions. They are the atomic data type, i.e. they cannot be divided further

1.1.1.2 Non-Primitive Data Structures

These are data structures composed of or derived from primitive data structures.

1.1.1.3 Linear Data Structures

These data structures are characterized by the fact their elements can be accessed in a sequential / linear manner. This means every element, excluding the first and last elements which only have a successor and predecessor respectively, has both a predecessor and successor, allowing bi-directional traversal.

1.1.1.4 Non-Linear Data Structures

These data structures are characterized by the fact that their elements are accessed based on some relationship / association between elements, therefore non-linearly.

1.2 Non-Primitive Data Structures

1.2.1 Arrays

> **Definition 1.2.1: Array**
>
> An indexed collection of a fixed number of homogeneous data elements. Elements are stored in contiguous memory, i.e. sequentially and can be accessed by their index.

1.3 Generics

> **Definition 1.3.1: Type Parameter**
>
> A placeholder for a type that is used in the definition of a generic type.

> **Definition 1.3.2: Generic Type**
>
> A type that is defined with one or more type parameters, i.e., `<T>`. These type parameters are used to define the type of the data that the generic type can store.