Block 1

Unit 1: Introduction to Data Structure

Structure

1.0 INTRODUCTION

This unit is an introductory unit and gives you an understanding of data structure, Algorithm, Data representation, various Data types and a general overview about linear and non-linear data structures. It is about structuring and organizing data as a fundamental aspect of developing a computer application.

1.1 Objectives

After the end of this unit, you should be able to:

1. Understand about algorithm.
2. Understand of the data organization and representation
3. Define the term data structure
4. Understand about various data types
5. Know the classification of data structure i.e. linear and non-linear
6. Introduce with data structure representation and operation on data structures

1.2 Algorithm definition

An algorithm is a set of instructions to be done sequentially. Any work to be done can be thought as series of steps. For example, to perform an experiment, one must do some sequential tasks like:

1. Set up the required apparatus
2. Do the process required
3. Note any observations
4. Summarize the results

These tasks accomplish an experiment. Let us see where such sequential steps are employed.

A computer or other electronic device which accomplishes a logical task is not actually logical but is simply following a series of programmed, sequential instructions. A computer algorithm consists of a series of well-defined steps given to the computer to follow. We can also define the algorithm as:

1. "An algorithm is a well define procedure that takes some value as input and produces some value as the output in finite number of steps."
2. "An algorithm is thus a sequence of computational steps that transform the input into the output that must halt after a final number of steps or time"
3. "An algorithm is a procedure for processing that is formulated so precisely that it may be performed by a mechanical or electronic devices must be formulated so exactly that the sequence of the processing steps is completely clear and it has to terminate in definite time."

The typical examples for algorithms are computer programs written in a formal programming language.

Example: Write an algorithm to exchange the value of two variables.

Algorithm:

1. ***Consider the two variables a, b with some values.***
2. ***Consider a third variable t.***

```
3. Do the following steps
        (a) Assign the value of a to t
        (b) Assign the value of b to a
        (c) Assign the value of t to b
4. Print the exchange value of a and b
```

1.3 Basic criteria for algorithms:

There are following basic criteria for an algorithm:

1. The algorithm must be expressed in a fashion that is completely free of ambiguity.
2. The algorithm must be efficient. It should not unnecessarily use memory locations nor should it require an excessive number of logical operations.
3. The algorithm should be concise and compact to facilitate verification of their correctness.
4. The algorithm must be independent from any programming language mostly its written in pseudo code, so it can convert to any programming language with the proper use of programming language syntax.

1.4 Data structure definition

First we define the meaning of simple data. Data are simply values or set of values. A data item is either the value of a variable or a constant. For example, the value of a variable x is 5 which is described by the data type integer, a data item is a row in a database table, which is described by a data type. A data item that does not have subordinate data items is called an elementary item. A data item that is composed of one or more subordinate data items is called a group item. A record can be either an elementary item or a group item. For example, an employee's name may be divided into three sub items – first name, middle name and last name but the social_security_number would normally be treated as a single item. Data may be organized in many different ways:

- The logical or mathematical model of a particular organization of data is called a data structure.
- A data structure is an arrangement of data in a computer's memory or even disk storage.

- Data structure is the method to store and organize data to facilitate access and modifications
- A data structure, sometimes called data type, can be thought of as a category of data. Like, Integer is a data category which can only contain integers. String is data category holding only strings. A data structure not only defines what elements it may contain, it also supports a set of operations on these elements, such as addition or multiplication.
- Data structures are ways to organize data (information) for example:
  - Simple variables are consider as the Primitive types
  - Array, the collection of data items of the same type, stored in memory at contiguously
  - Linked list, the sequence of data items, each one points to the next one, stored in memory at non-contiguously.
- Data structures are building blocks of a program. If program is built using improper data structures, then the program may not work as expected always.
- The possible ways in which the data items are logically related define different data structures.
- A data structure is a collection of different data items that are stored together in a clearly defined way.

The examples of several common data structures are **string, arrays, Stacks, Queues, Linked list, Binary Trees, Graph and Hash Tables.**

In combination with Algorithm we may define the data structures as:

- Algorithms go with the data structures to manipulate the data i.e. Algorithms are used to manipulate the data contained in these data structures as in the form of sorting and searching.
- More generally we can say: **Algorithms + Data Structures = Programs.**

1.5 Data Type

A data type is a classification of data, which can store a specific type of information. Data types are primarily used in computer programming, in which variables are created to store data. Each variable is assigned a data type that determines what type of data the variable may contain. Thus a data type is a method of interpreting a pattern of bits. There are numerous different data types. They are used to make the storage and processing of data easier and more efficient.

A data type is a term which refers to the kinds of data that variables may hold. With every programming language there is a set of built-in data types. This means that the language allows variables to name data of that type and provides a set of operations which meaningfully manipulates these variables. Some data types are easy to provide because they are built-in into the computer's machine language instruction set, such as integer, character etc. Other data types require considerably more efficient ways to implement. In some languages, these are features which allow one to construct combinations of the built-in types (like structures in 'C'). However, it is necessary to have such mechanism to create the new complex data types which are not provided by the programming language. The new type also must be meaningful for manipulations. Such meaningful data types are referred as abstract data type.

Different programming languages have their own set of basic data types.

Basic data types or primitive data types

The most common basic or intrinsic data types or primitive data types are as follows:

- Integer: It is a positive or negative number that does not contain any fractional part.
- Real: A number that contains the decimal part
- Boolean: It is a data type that can store one of only two values, usually these values are TRUE or FALSE
- Character: It is any letter, number, punctuation mark or space, which takes up a single unit of storage, usually a byte
- String: It is sometimes just referred to as 'text'. Any type of alphabetic or numeric data can be stored as a string: "Delhi City", "30/05/2013" and "459.78" are all examples of the strings. Each character within a string will be stored in one byte using its ASCII code. The maximum length of a string is limited only by the available memory.

Structure data types or Non Primitive data types

There is another class of data types which is considered as structure data types or non primitive data types. These data types are user defined data types. Structured data types hold a collection of data values. This collection will generally consist of the primitive data types. Examples of this would include arrays, records, list, tree and files. These data types, which are created by programmers, are extremely important and are the building block of data structures. These are more complex data structures. They stress on formation of sets of homogeneous and heterogeneous data elements.

Abstract data types or Non-primitive data types

It is another form of the non-primitive data types. An abstract data type can be assumed as a mathematical model with a collection of operations defined on that model i.e. an Abstract data type (ADT) is a new data type derived or created from basic or built in data type based on a particular logical or mathematical model. For example Set of integers consisting of different numbers may be an ADT. A set is a combination of more than one integer, but the operations on set is a generalized operation of different integers such as union, intersection, product, and difference.

Note: If the data contains a single value this can be organized using primitive data type. If the data contains set of values they can be represented using non-primitive data types.

Now on the basis of above mentioned data types the data structure can be defined as:

"*An implementation of abstract data type is data structure i.e. a mathematical or logical model of a particular organization of data is called data structure.*"

1.6 Type of Data Structures

A data structure is the portion of memory allotted for a model, in which the required data can be arranged in a proper fashion. Normally The data structures are of two types or it can be broadly classified into two types of data structures:

(i) Primitive data structure

(ii) Non-primitive data structure

1. Primitive data structure: The data structures that typically are directly operated upon by machine level instruction. Examples: Integers, Real numbers, Characters and pointers, and their corresponding storage representation. Programmers can use these data types when creating variables in their programs. For example, a programmer may create a variable say "z" and define it as a real data type. The variable will then store data as a real number.

2. Non primitive data structure: Non – primitive data structures are not defined by the programming language, but are instead created by the programmer. They are also called as the reference variables, since they reference a memory location, which stores the data. These data structures are derived from the primitive data structures. Examples: Array, Stack, Queues, Linked list, Tree, Graphs and hash table.

There are two type of-primitive data structures.

a) Linear Data Structures:-

In linear data structure the elements are stored in sequential order. Hence they are in the form of a list, which shows the relationship of adjacency between elements and is said to be linear data structure. The most, simplest linear data structure is a 1-D array, but because of its deficiency, list is frequently used for different kinds of data. The linear data structures are:

(i) **Array:** The Array is a collection of data of same data type stored in consecutive memory location and is referred by common name.

(ii) **Stack:** A stack is a Last-In-First-Out or First-In-Last-Out linear data structure in which insertion and deletion takes place at only from one end called the top of the stack.

(iii) **Queue:** A Queue is a First-In-First-Out or Last-In-Last-Out data structure in which insertion takes place from one end called the rear and the deletions takes place at one end called the Front.

(iv) **Linked List:** Linked list is a collection of data of same type but the data items need not be stored in consecutive memory locations. It is linear but non-contiguous type data structure. A linked list may be a single list or double list.

- **Single Linked list: -** A single list is used to traverse among the nodes in one direction.
- **Double linked list: -** A double linked list is used to traverse among the nodes in both the directions.

b) Non-linear data structure:-

In non linear data structure the elements are stored based on the hierarchical relationship among the data. A list, which doesn’t show the relationship of adjacency between elements, is said to be non-linear data structure. The non-linear data structures are:

(i) **Tree:** This data structure is used to represent data that has some hierarchical relationship among the data elements. Thus, it maintains hierarchical relationship between various elements.

(ii) **Graph:** This data structure is used to represent data that has relationship between pair of elements not necessarily hierarchical in nature. It maintains random relationship or point-to-point relationship between various elements. For example electrical and communication networks, airline routes, flow chart and graphs for planning projects.

1.7 REPRESENATION OF DATA STRUCTURES

There are generally two common methods for the data structure representation. These methods can be specified as:

(i) Sequential representation

(ii) Linked representation

(i) Sequential representation

A sequential representation maintains the data in continuous memory locations which takes less time to retrieve the data but leads to more time during insertion and deletion operations due to its sequential nature.

(ii) Linked Representation

Linked representation maintains the list by means of a link between the adjacent elements which need not be stored in continuous memory locations. During insertion and deletion operations, links will be created or removed between which it takes less time when compared to the corresponding operations of sequential representation. Generally, linked representation is preferred for any data structure.

1.8 Data structure Operations

The data elements appearing in the data structure is processed by means of certain operations. In fact the particular data structure that one chooses for a given situation depends largely on the frequency with which specific operations are performed. The following major operations performed on data structures are:

Insertion

It provides the means for adding new details or new node into the existing data structure.

Deletion

It provides the means for removing a node from the data structure.

Traversing

It provides the means for accessing each node exactly once so that the nodes of a data structure can be processed. It is also called the visiting to data structure.

Searching

It provides the means for finding the location of node for a given key value or finding the locations of all records, which satisfy one or more conditions.

Sorting

It provides the means for arranging the data in a particular order in given data structure.

Merging

It provides the means for joining the two data structures.

Note: Sometimes two or more data structure of operations may be used in a given situations; e.g. we may want to delete the records with a given key, which may means we first need to search for the location of the record.

1.9 Summary

- Data structure is the particular organization of data either in a logical or mathematical manner
- Data type is a concept that defines internal representation of data.
- Data structures are building blocks of a program. If program is built using improper data structures, then the program may not work as expected always
- An algorithm is a set of instructions to be done sequentially. Any work to be done can be thought as series of steps.

- An abstract data type is the specification of logical and mathematical properties of data types or structure. It acts as a guideline to implement a data structure.
- A data structure is the portion of memory allotted for a model, in which the required data can be arranged in a proper fashion
- Primitive and non-primitive are the two basic data types of data structure.
- The relationship between abstract data type and data structure is well defined. An abstract data type is the specification of a data type whereas data type is the implementation of abstract data type and data structure comprises computer variable of same or different data types.
- There are generally two common methods for the data structure representation i.e. Sequential and linked representation.

Bibliography

Horowitz, E., S. Sahni: “Fundamental of computer Algorithms”, Computer Science Press, 1978

J. P. Tremblay, P. G. Sorenson “An Introduction to Data Structures with Applications”, Tata McGraw-Hill, 1984

M. Allen Weiss: “Data structures and Problem solving using C++”, Pearson Addison Wesley, 2003

Ulrich Klehmet: “Introduction to Data Structures and Algorithms”, URL: http://www7 . Informatik.uni-erlangen.de/~klehmet/teaching/SoSem/dsa

Markus Blaner: “Introduction to Algorithms and Data Structures”, Saarland University, 2011

R. B. Patel “Fundamental of Data Structures in C”, PHI Publication

V. Abo. Hopcropft, Ullaman, “data Structure and Algorithms”, I.P.E.

Seymour Lipschutz, “Data Structure”, Schaum’s outline Series.

Self Evaluation

1. Define the data structure and Algorithm.
2. What are different data types? Give the example of each data type.
3. Specify the types of data structure with the example of each type.
4. What are the various operations on data structures?
5. While Considering the data structure implementations, the factor under consideration is/are:
    a. Time
    b. Space and Time
    c. Time, Space and Processor
    d. None of the above
6. A data type is the collection of values and the set of operations on values (True/False)
7. ………………..refers to the collection of computer variables that are connected in some specific manner.
8. One of the example of a structured data type can be…………………..
9. Explain abstract data type with an example.
10. What is a data structure and what are the difference between data types, abstract data type and data structure?
11. An -------------data type is a keyword of a programming language that specifies the amount of memory needed to store data and the kind of data that will be stored in that memory location.

a. Abstract b. int c. vector d. None of these

12. Graphs are classified into ……………. category of data structure.
13. What do you mean by LIFO and FIFO?