FUNDAMENTALS OF DATA STRUCTURES

Introduction:

What is Data? In Computer Science, Data is any representation of a fact that can

be communicated or manipulated by some process.

Also Data can be defined as " Input to à computer, processed into information

under the direction of a program"

Examples are

- Number of hours worked

- Some level of stock

DATA STRUCTURES:

A Data Structure is a construct used to organize information to make it easy and

efficient to access and process.

Other definions are;

- Data structure is defined as a particular way of storing and organizing data in

computer so that it can be used efficiently.

- Data structure refers to the organization of data in computer memory or the

way in which data is efficiently stored, processed and retrieved

- A data structure is a way of storing data in a computer so that it can be used

efficiently and it will allow the most efficient algorithm to be used.

-

Data structure is a structural representation of logical relationships between

elements of data.

- Data structure is defined as a way of representing data in a computer memory

when complex data involving many types are present.

Data structures are generally based on the ability of a computer to fetch and

store data at any space in its memory specified in an address.

Classification of Data Structures:

Data structures can be classified as

- Simple data structure

El Compound data structure

I Linear data structure

m Non linear data structure

Operations applied on linear data structure:

The following is a list of operations applied on linear data structures

1. Add an element

2. Delete an element

3. Traverse

4. Sort the list of elements

5. Search for a data element

For example Stack, Queue, Tables, List, and Linked Lists.

Non-linear Data Structure:

Non-linear data structure can he constructed as a collection of randomly

distributed set of data items joined together by using a special pointer (tag). In

non-linear Data structure the relationship of

adjacency is not maintained between the data items,

Operations applied on non-linear data structures;

The following is a list of operations applied on non-linear data structures.

1. Add elements

2. Delete elements

3. Display the elements

4. Sort the list of elements

5. Search for a data element

For example Tree, Decision tree, Graph and Forest

Abstract Data Type;

An abstract data type, sometimes abbreviated ADT, is a logical description of how

we view the data and the operations that are allowed without regard to how they

will be implemented. This means that we are concerned only with what data is

representing and not with how it will eventually be constructed, By providing this

level of abstraction, we are creating an encapsulation around the data.

The idea is that by encapsulating the details of the implementation, we are hiding

them from the user's view.

When you solve a problem with a computer program, always ask first,

"How should the program store the information upon which it computes?"

For example, If the program is a spreadsheet program, then the information

should be held in a data structure that is a grid.

If the program is a bank-account database, then the information should be

grouped into customer accounts, each with a unique ID, saved in an array or set.

If the program is a file-system manager, then the information are files and folders

that are organized in a tree-like structure.

Each of these problems required a distinet data structure in the solution.

The purpose of a data-structures course like CMP212 is to train you at using a

variety of such structures.

Further:

A data structure has methods for

inserting an element

retrieving an element

deleting an element

APPLICATIONS OF DATA STRUCTURES

1. Compiler design

2. Data management system

3. Simulation

4. Operating system

Data Structures are very useful in the development of software.

Specific data structures are essential ingredients of many algorithms and it makes

possible the management of huge amount of data such as large databases and

internet indexing services.

Types of Data Structure

1. Array

2. Ordered array

3. Stack

4 Queue

5. Linked list

6. Binary Trees

7. Hash Table

8. Heap

9. Graph

For many problems, the ability to formulate an efficient algorithm depends on

being able to organize the data in an appropriate manner.

A data structure should be seen as a logical concept that must address two

fundamental concerns,

1. First, how the data will be stored, and

2. Second, what operations will be performed on it.

FUNDAMENTALS OF DATA STRUCTURES-2

Data Types:

All data is represented in the computer by strings of bits. It follows that from the

bits themselves we cannot determine the value of stored quantity. For example,

the bits 10111100 might represent the integer188 or the character U. We can

only interprete the value unambiguously if we associate a type with each item of

stored data, Hence in programming types impose some constraints enforcing

correctness so that the ambiguity is removed.

Associated with each type, there is a set of values. In our example above, 188 is a

value of type integer, and U is a value of type character. So for a programming

language, 188 and U are entities of different types and are therefore unrelated,

even though they happen to have the same representation (10111100 ) in our

computer.

In addition to the set of values, we associate with each type literal, constants,

variables and operators. A literal is simply a value of a particular type; for

example, 10 is an integer literal, and 'H' is a character literal. Notice the

apostrophes (qoutes) because we want to use the symbol + for addition. It is a

convention to write character literals between apostrophes, Also note that O is

an integer literal, but 'O' is a character literal.

A constant is a name that is a synonym for a literal.

A variable has name, a type, and a value that belongs to the set of values of its

type. The value that a variable assumes may change from time to time while the

program is running, but its type neva changes.

A data type is a way to represent a particular set of values and to determine what

operations can be perforned on those values.

Data Type Classification:

The simple data types are:

(a) Integers

(b) Real Numbers

(c) Characters

(d) Booleans

Integers are simple numbers that do not contain fractional parts such as 5, -52,

and 3,422.

Real numbers are numbers that include fractional parts such as 7.5, 3.667, -5.0,

00034.

Characters: All of the symbols that can be produced by pressing keys on a

keyboard. Characters are used to generate human readable text such as English

sentences. The Data Type Char is used to represent characters.

Booleans: The values false and true.

Primitive Data Types

Every computer language supports a set of native data types.

These types may be nat,ve (built-in) either to the machine on which the programs

are run or to the compiler or interpreter that is translating these programs.

Fixed-Point Representati»n of Real Numbers

The usual way of representing real numbers is to write the number with the

decimal point fixed in its correct position between the two appropriate digits,e.g., 13.75 or 3862.4. This is very useful in data processing for example where

sums of money are to be processed or printed.

However, this representation becomes laborous and cumbersome when dealing

with several large or very small numbers e.g. 1375000000, 386240000,

0.000001375, 0.0000018. The answer is to use the floating- point representation.

Floating-Point Notation

There are many variations of floating-point notation, each with its own individual

characteristics.

The key concept is that a real number is represented by a number, called a

MANTISSA, times a BASE raised to an integer power, called an EXPONENT. The

base is usually fixed and the mantissa and exponent vary to represent different

real numbers.

For example, if the base is fixed at 10, the number 387.53 could be represented

as 38,753 times 10 to the -2 power. The mantissa is 38753 and the exponent is -2.

Other possible representations are 38753 X103 and 387.53 \* 10°

The floating-point representation of 640,000 is 6,4 \* 10°. The mantissa is

6.4, the exponent 5 shows the power of 10 to which 6.4 is raised.

SCIENTIFIC NOTATION

The scientific notation is a floating-point method of representing a number,

especially a very large number or very small one, in which numbers are expressed

as products consisting of a number between 1 and 10 multiplied by a power of

10.

Scientific notation cormonly uses the letter E in place of "times 10", as in 5.0E3,

meaning 5.0 times 10 to the third power, or 10\*

Thus 640.000 = 6.4E+05 or 6.4ES.

Further Examples:

Fixed Point Representation

Floating -Point Representation

13.75

137.5

1375.

1375000000.

1.375

0.1375

0.01375

0.001375

3862.4

386240000.

.00000038624

For 1,375 \* 103

1.375 \* 101

1.375 \* 102

1.375 \* 103

1.375 \* 10°

1.375 \* 10°

1.375 \* 101

1.375 \* 102

1.375 \* 10-3

3.8624 \* 103

3.8624 \* 108

3.8624 \* 107

1.375 is the Mantissa

10 is the base or radix

3 is the exponent

FUNDAMENTALS OF DATA STRUCTURES-4

ARRAYS:

We can define array as follows,

An array is a group of objects referenced with tha same variable name. The

individual values in an array are elamen, Array elements are also variables. You

"dimension" an array when you use it the first time or ine Each element in array

is referred to by an array Variable subscripted with an integer or an integer

expression.

An array is a collection of variables of the same type Mat are referreu to

through a common name.

An array A is a set of items, of the sane type, which are so arranged that an

ordered set of integers unicuely defines 1 re position of each item of the array and

provides a method of accessing each iterdirectly. Each individual integer in the

ordered set is called a subscrin!.

Dimension Of an Array; If the numpy of elements in the ordered set of

subscripts is N, then the arrays said to be of dimension N. So when there is only

one subscript, the array is said to be ane-dimensional, When there are two

subscripts, the array is said to be tv, y dimensional when. there are three

subscripts, the array is said to be three-inensional. Generally, when there are

more than ONE subscripts, the array is sal! to be a multi dimensional array.

in programming languages, all arrays corsist of cor iguous Mcmul,

The lowest address corresponds to the hast element and the highest address to

the last element: Arrays may have from eye to severa dimensions,

Arrays store collections of data. Array is t'e most com non data structure

used to store collections ofelenents. Arr vs are convenient to declare and

provide the easy syntax to an cess any clerint by its index number.