SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name: 23ITB201 Data structures and Algorithms

II Year / III semester

Unit I – Abstract Data types

Topic: Abstract Data types

ABSTRACT DATA TYPES

Introduction to Abstract Data Types (ADTs)

ADT of Mobile phone

Abstract View:

- Ram capacity
- Processor Speed
- LCD screen size
- Dual Camera
- Android

Operations:

- Call()
- Messaging()
- Photo()
- Video()

Definition on ADT

- The abstract data types are the entities that are definitions of data and operations but do not have implementation details.
- The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented.
- It is called “abstract” because it gives an implementation-independent view.

Abstract Data type

Various data structures that can be used for ADT are

- Array
- Linked list
- Stack
- Queue
- Trees
- Graphs etc.,

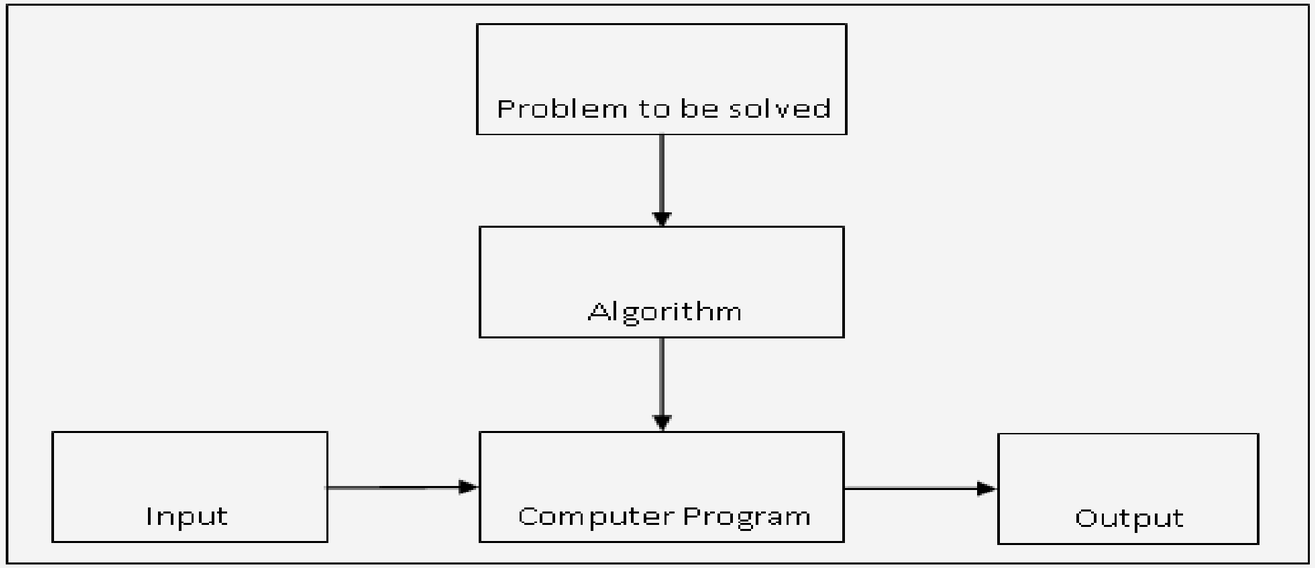
ASSESSMENT – 1

- **What is abstract data type?**
- **ADT of smart watch**

Introduction to analysis of algorithms

NOTION OF AN ALGORITHM

- An ***algorithm*** is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.
- It is a step by step procedure with the input to solve the problem in a finite amount of time to obtain the required output.



Introduction to analysis of algorithms

The notion of the algorithm illustrates some important points:

- The **non-ambiguity requirement** for each step of an algorithm cannot be compromised.
- The **range of inputs** for which an algorithm works has to be specified carefully.
- The same algorithm can be represented in several different ways.
- There may exist several algorithms for solving the same problem.
- Algorithms for the same problem can be based on very different ideas and can solve the problem with dramatically **different speeds.**

Introduction to analysis of algorithms

Characteristics of an algorithm:

Input: Zero / more quantities are externally supplied.

Output: At least one quantity is produced.

Definiteness: Each instruction is clear and unambiguous.

Finiteness: If the instructions of an algorithm is traced then for all cases the algorithm must terminates after a finite number of steps.

Efficiency: Every instruction must be very basic and runs in short time.

Introduction to analysis of algorithms

Steps for writing an algorithm:

1. An algorithm is a procedure. It has two parts; the first part is head and the second part is body.

2. The Head section consists of keyword Algorithm and Name of the algorithm with parameter list. E.g. Algorithm name1(p1, p2,…,p3)

The head section also has the following:

//Problem Description:

//Input:

//Output:

3. In the body of an algorithm various programming constructs like if, for, while and some statements like assignments are used.

4. The compound statements may be enclosed with { and } brackets. if, for, while can be closed by endif, endfor, endwhile respectively. Proper indention is must for block.

Introduction to analysis of algorithms

5. Comments are written using // at the beginning.
6. The identifier should begin by a letter and not by digit. It contains alpha numeric letters after first letter. No need to mention data types.
7. The left arrow "←" used as assignment operator. E.g. v←10
8. Boolean operators (TRUE, FALSE), Logical operators (AND, OR, NOT) and Relational operators (<,<=, >, >=,=, ≠, <>) are also used.
9. Input and Output can be done using read and write.
10. Array[], if then else condition, branch and loop can be also used in algorithm.

Introduction to analysis of algorithms

Example:

The greatest common divisor(GCD) of two nonnegative integers m and n (not-both-zero), denoted gcd(m, n), is defined as the largest integer that divides both m and n evenly, i.e., with a remainder of zero.

Euclid's algorithm for computing gcd(m, n) in simple steps

Step 1 If n = 0, return the value of m as the answer and stop; otherwise, proceed to Step 2.

Step 2 Divide m by n and assign the value of the remainder to r.

Step 3 Assign the value of n to m and the value of r to n. Go to Step 1.

Introduction to analysis of algorithms

Euclid’s algorithm for computing gcd(m, n) expressed in pseudocode

ALGORITHM Euclid_gcd(m, n)

//Computes gcd(m, n) by Euclid’s algorithm

//Input: Two nonnegative, not-both-zero integers m and n

//Output: Greatest common divisor of m and n

while n ≠ 0 do

r ←m mod n m←n

n←r

return m

ASSESSMENT – 2

- What are the properties of an algorithm?
- List the steps involved in writing an algorithm?