11-2-26

Q What is algorithm specification? Explain its characteristics. How to write an algorithm? Give example.

Ans→ Algorithm is a step-by-step procedure which defines a set of instructions to be executed in a certain order to get the desired result (output).

Algorithms are generally created independent of underlying languages.

Characteristics of an algorithm :-

An algorithm should have the following characteristics —

Unambiguous — Algorithm should be clear and each of its steps and their inputs/outputs should be clear and must lead to only one meaning.

Input — An algorithm should have 0 or more well-defined inputs.

Output — An algorithm should have 1 or more well-defined outputs and should match the desired output.

Finiteness — Algorithms must terminate after a finite number of steps.

Feasibility — Algorithms should be feasible with the available resources.

Independent — An algorithm should have step-by-step directions, which should be independent of any programming code.

How to write an algorithm →

Ex:- To add two numbers and display the result.

Step 1 — START
Step 2 — Declare three integers a, b and c.
Step 3 — Define values of a and b.
Step 4 — Add values of a and b.
Step 5 — Store output of step 4 to c
Step 6 — Print c
Step 7 — STOP

Alternatively it can also be written as —

Step 1 — START ADD
Step 2 — get values of a and b
step 3 — c ← a + b
step 4 — display c
step 5 — STOP

Algorithm Complexity :—

Suppose X is an algorithm and n is the size of input data, the time and space used by the algorithm X are the two main factors, which decide the efficiency of X.

The complexity of an algorithm f(n) gives the running time and/or the storage space required by the algorithm in terms of n as the size of input data.

Space Complexity — Space complexity of an algorithm represents the amount of memory space required by the algorithm in its life cycle. The space required by an algorithm is equal to the sum of the following two components —

→ A fixed part- space required to store certain data types and variables. For example — variables, constants, program size etc.

→ A variable part space required by variables, whose size depends on the size of the requirement (problem). Ex — dynamic memory allocation recursion stack space etc.

Space Complexity S(P) of an algorithm P is

$S(P) = C + SP(I)$ where C is the fixed part.
SP(I) is the variable part of the algorithm

Eg:- Algorithm : SUM (A, B)

Step 1 — START

Step 2 — C ← A + B + 10

Step 3 — STOP

Here A, B, C are three variables, 10 is a constant value.

∴ S(P) = 1 + 3 = 4

Now, Space depends on data type of given variables and constant types and it will be multiplied accordingly.

Time Complexity — Time complexity of an algorithm represents the amount of time required by the algorithm to run to completion. Time requirements can be defined as a numerical function T(n), where T(n) can be measured as the number of steps provided each step consumes constant time.

For example, addition of two n-bit integers takes n steps.

Consequently, the total computational time is T(n) = C * n, where C is the time taken for the addition of two bits. Here, we observe that T(n) grows linearly as the input size increases.