

ANALYSIS OF ALGORITHM

An algorithm is a step by step method of solving a given problem.

OR

An algorithm is a finite set of instructions that, if followed, accomplishes a particular task.

CHARACTERISTIC OF ALGORITHMS

Algorithms generally have the following characteristics :

1. **INPUT** : The algorithm receives input. Zero or more quantities are externally supplied.
2. **OUTPUT** : The algorithm produces output. Atleast one quantity is produced.
3. **PRECISION** : The steps are precisely stated. Each instructions is clear and unambiguous.
4. **DETERMINISM** : The intermediate results of each step of execution are unique and are determined only by the inputs and the results of the precedings steps.
5. **FINITENESS** : If we trace out the instructions of an algorithm, then for all cases the algorithm terminates after a finite number of steps.
6. **CORRECTNESS** : The output produced by the algorithm must be correct i.e. algorithm correctly solves the problem.
7. **GENERALITY** : The algorithm applies to a set of inputs.
8. **EFFECTIVENESS** : Every instruction must be very basic so that it can be carried out, in principle, by a person using only pen and paper.

Example : 1

The following algorithm find the minimum of three numbers x , y and z .

1. $\text{Min} = x$
2. If $y < \text{Min}$ then $\text{Min} = y$
3. If $z < \text{Min}$, then $\text{Min} = z$

At the end of algorithm, Min will be equal to the minimum of three numbers.

ANALYSIS (COMPLEXITY) OF ALGORITHMS

The analysis of an algorithm refers to the process of deriving estimates for the time and space required to execute the algorithm.

It is important to estimate time (that is number of steps) and space (e.g. the number of variables) required by algorithms.

Knowing the time and space required by algorithms allows us to compare algorithms that solve the same problem.

The example if an algorithm taken ' n ' steps to solve a problem and another algorithm take n^2 steps to solve the same problem. We would obviously prefer the first algorithm. Thus estimation of time and space needed to execute the algorithm is called the time and space complexity of the algorithm.

The time needed to execute an algorithm is a function of the input. There are three cases worth nothing about the time complexity of an algorithm is a different task.

Best Case Time : The best case time of an algorithm is the minimum number of steps that can be executed for the given parameters.

Worst Case Time : The worst case time of an algorithm is the maximum number of steps that can be executed for the given parameters.

Average Case Time : The average case time of an algorithm is the average number of steps that can be executed for the given parameters.

We are mainly concerned with estimating the time of an algorithm rather than computing its exact time.

Example :

Consider that the worst case time of an algorithm is $f(n) = 20n^2 + 4n + 5$

For some input of size n . Then for large n , the terms $20n^2$ is approximately equal to $f(n)$. (This can be checked by putting the values of n). Thus it can be claimed that $f(n)$ grows like $20n^2$.