

ALGORITHMS

The word Algorithm means
“A set of rules to be followed
in calculations or other problem
- solving operations”.

It means that Algorithm is
just a step - by - step procedure
to solve a problem.

Characteristics of an Algorithm

(i) Clear & Unambiguous :-

Each of its steps should be clear in all aspects & must lead to only one meaning.

(iii) Well defined inputs &
Outputs:-

If the Algorithm takes an inputs that it should be well defined. Similarly the outputs should be well defined.

It may or may not take inputs But, It should produce at least one output.

(iii) Finite-ness :-

The algorithm should terminate after a finite time.

(iv) Feasible:-

The algorithm must be simple, generic & practical such that it can be executed using available resources.

(v) Language Independence:-

It should be made in such a way it can be done in any language by following instructions.

Types of Algorithms :-

(i) Brute Force Algorithms:-

It is the simplest approach to a problem.

Ex. You have a padlock with 3 digits each from 0 to 9. Then you will have to check all 1000 combinations from 000 to 999 to get the solution.

(ii) Sorting Algorithms:-

Sorting is arranging the

group of data in a particular manner according to requirement.

Example :-

- (i) Bubble Sort
- (ii) Selection Sort
- (iii) Insertion Sort
- (iv) Merge Sort
- (v) Quick Sort

(iii) Hashing Algorithms:-

They are used for searching elements but every element

is stored in such a way
that index holds the Key id.

Example:-

(i) Division Method

(ii) Mid-Square Method

(iii) Folding Method

(iv) Multiplication Method

(iv) Recursive Algorithms:-

A problem broken into
several subproblems & we
call the same function again.

Example :-

⇒ Fibonacci Series

$$F(n) = F(n-1) + F(n-2)$$

(V) Searching Algorithms:-

These Algorithms help in finding elements with a specific value inside data structures.

Example :-

- ij) Linear Search
- iii) Binary Search

(vi) Divide & Conquer Algorithms

This algorithm breaks a problem into sub-problems, solves a single subproblem & merge them to get a final solution.

It has three steps:-

- (i) Divide
- (ii) Solve
- (iii) Merge or Combine

Example :-

- (i) Binary Search
- (iii) Merge Sort

(Vii) Greedy Algorithms :-

It makes the locally optimal choices at each step with the hope of finding the globally optimal solution.

Examples :-

Huffman Coding

(Viii) Backtracking Algorithm

Backtracking is a problem solving algorithmic technique that involves finding a solution incrementally

by trying different options & undoing them if we reach a dead end.

Types of Backtracking Problem :-

(i) Decision Problems :-

In this we search for a feasible solution.

(ii) Optimization Problem:-

We search for the best solution in this.

(iii) Enumeration Problem:-

We find set of all possible solutions to the problem of this type.

Example:-

(i) Sudoku

(ii) Puzzles

(ix) Dynamic Programming

DP is a method used to solve complex problems by breaking

them down into simpler subproblems. By solving each subproblem only once & storing the results, it avoids redundant computations, leading to efficient solutions.

Types when to use DP:-

(i) Optimal Substructure:-
Optimal Substructure means that we combine the

optimal results of subproblems to achieve optimal result of the bigger problem.

Example :-

Minimum cost path in a weighted graph.

(iii) Overlapping Subproblem

The same subproblem are solved repeatedly in different parts of problem.

Example :- Fibonacci Series

Two approaches:-

- (i) Top-Down (Memoization)
- (ii) BottomUp (Tabulation)

(X) Randomized Algorithm

These are algorithms that use randomness in their computations to achieve a desired outcome. These algorithms introduce randomness to improve

efficiency. It is used where exact solutions are difficult to find or when probabilistic approach is acceptable.