Algorithms analysis and design :->

Syllabus :-

1. Analysis
2. Divide and conquer
3. Greedy techniques
4. Dynamic programming
5. Hashing – Tree – Graph.

Algorithms

```
Ex.  ATM ()
     {
        Take 2- No. (a, b)
        c = add (a, b)
        Print (c)
     }
```

1. It should terminate after finite time.
2. It should produce at least one output.
3. It is independent of programming language.
4. It is deterministic in the nature.

Steps required to design algorithm.

1. Problem definition (Knowing problem clearly
2. Design algorithm ->
   - DAC (Divide and conquer)
   - Greedy technique
   - Dynamic programming
   - Backtracking

   etc
3. flow-chart
4. Verification
5. Coding
6. Analysis.

- Analysis
  - To solve a perticular problem, if at all more than one algorithm is exist,

    Best one will be decided by Analysis based on two parameters.

    I) Time (CPU Time)

    II) Space (Main memory space)

- Time Complexity (CPU Time) :->

$$T(P) = C(P) + R(P)$$

$C(P)$ → Compiler → Software → Programming language of compiler.

$R(P)$ → CPU → Hardware → Types of hardware

Both → Analysis

- Analysis :→

  1. Posteriori analysis
  2. Priori analysis

Posteriori analysis	Priori analysis
• Relative analysis	• Absolute analysis
• It is dependent on programming language of compiler and type of hardware.	• It is independent of programming language of the compiler and type of hardware.
• Exact answer.	• Approx answer.
• Computer to computer answer is changing.	• Same answer in every computer.

- apriori analysis :→
  - It is a determination of order of magnitude of a statement.

→ while running, no. of time statements will execute.

+, − ⇒ O(n)	^, √ ⇒ O(log log n)
*, / ⇒ O(log n)	