

# II. Algorithms analysis and design :-

Syllabus :-

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

## 0 Algorithms

Ex ATN()

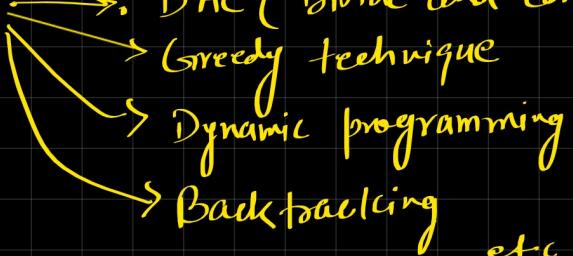
{  
    Take 2- No. (a, b)  
    C = add (a, b)  
    Print (c)

}

- 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)



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

## • Analysis

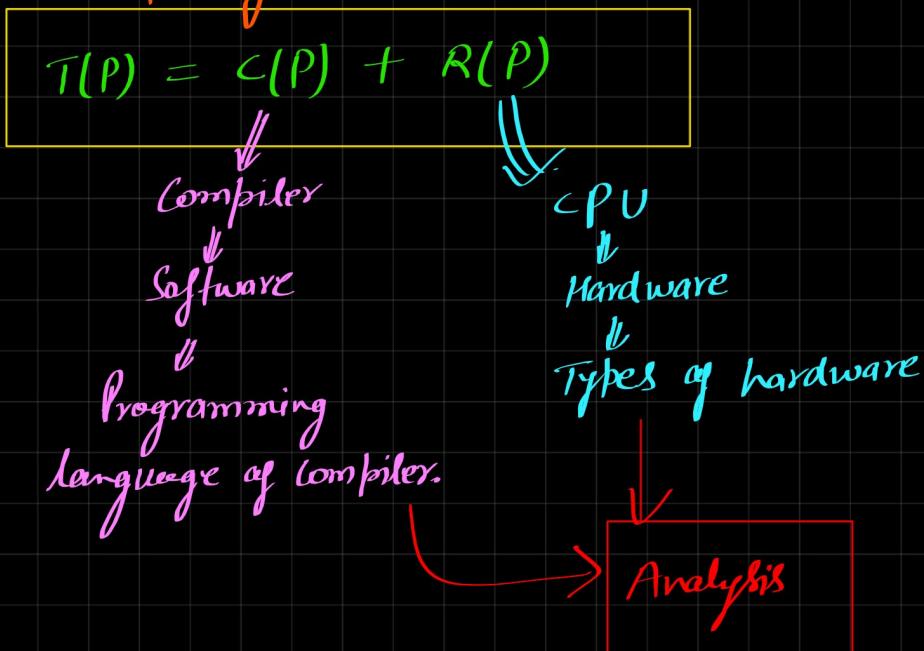
- To solve a particular problem, if at all more than one algorithm is exist,

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

I) Time ( CPU Time )

II) Space ( Main memory space )

## • Time Complexity ( CPU Time ) $\Rightarrow$



## o Analysis $\rightarrow$

$\nexists$ . Posteriori analysis  
 $\exists$ . Priori analysis

### Posteriori analysis

- Relative analysis
- It is dependent on programming language of compiler and type of hardware.
- Exact answer.
- Computer to computer answer is changing.

### Priori analysis

- Absolute analysis
- It is independent of programming language of the compiler and type of hardware.
- Approx answer.
- Same answer in every computer.

## o apriori analysis $\rightarrow$

- It is a determination of order of magnitude of a statement.

while running, no. of time statements will execute.

$$\begin{array}{|c|} \hline + \\ \hline - \\ \hline \end{array} \Rightarrow O(n)$$

$$\begin{array}{|c|} \hline * \\ \hline / \\ \hline \end{array} \Rightarrow O(\log n)$$

$$\begin{array}{|c|} \hline \wedge \\ \hline \sqrt{} \\ \hline \end{array} \Rightarrow O(\log \log n)$$









