Design and Analysis of Algorithms

Introduction to Algorithms

Books To Be Referred

- Fundamentals of Computer Algorithms –
  - Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajakaran
  - Galgotia Publications

- Introduction to Algorithms , IInd Edition –
  - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
  - Prentice – Hall India (PHI)

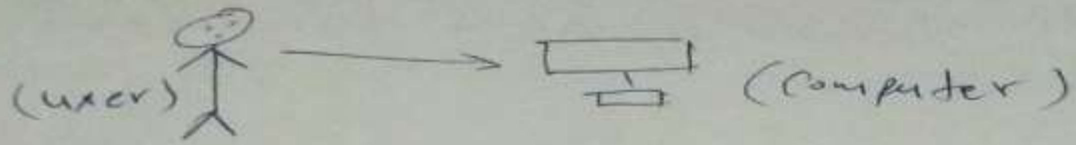
Content

- What is Algorithm?
- Why Study Algorithm?
- Algorithm Specifications
- Analysis of Algorithms
- Algorithms Design Strategies/Techniques

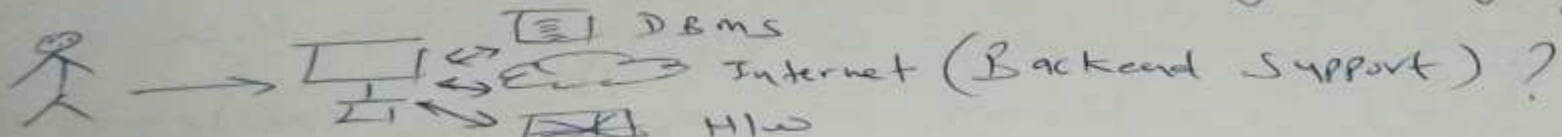
What is Algorithm?

Introduction to Algorithm →

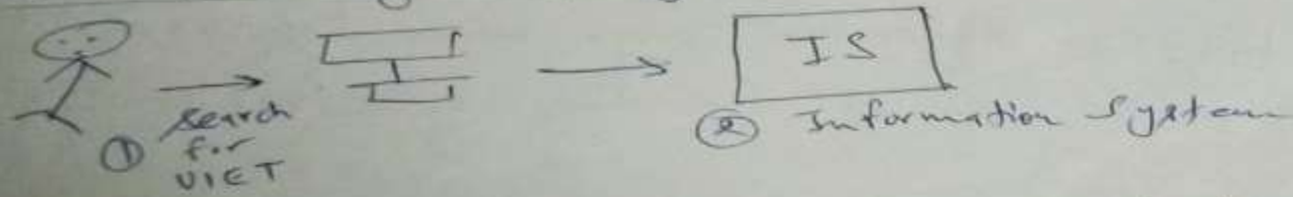
– Introduction →



A man thinks a Computer can do any thing & everything



Scenario for searching a query →



① search → {A Common man rarely understands that a man made procedure called search has done the entire job; But It's not real Behind this

② Information System → It should know what user can frequently search i.e → VIET.

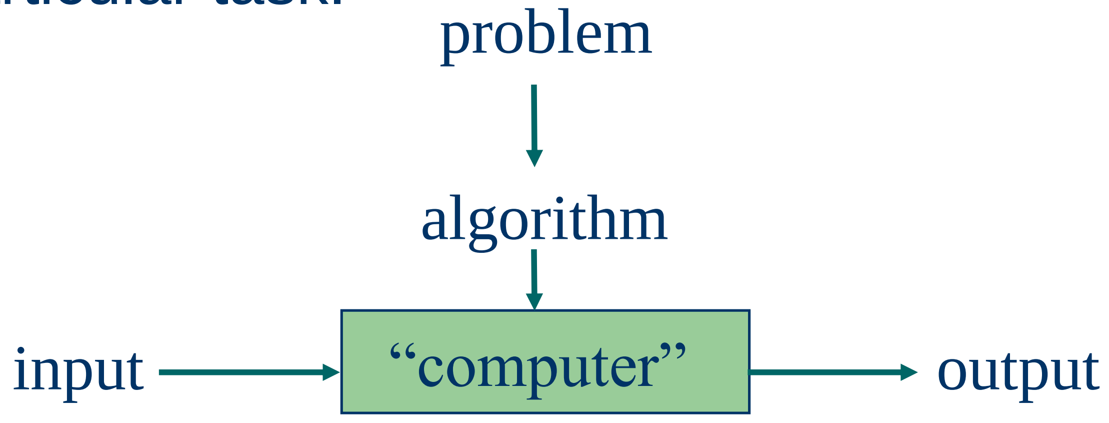
$IS \xrightarrow{\text{make}}$ structured Information $\xrightarrow{\text{store}}$ Computer → Memory

$\xrightarrow{\text{~~Requirement~~}}$ Right Info. $\xrightarrow{\text{accomplished}}$ A set of Instructions / Program
↓ Created by
Designer of IS

[Program → Algo. ?]

Algorithm

- An *algorithm* is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.
- A finite set of instructions which if followed accomplish a particular task.



Definitions of Algorithm →

Name Algorithm Comes from (Persian) Afghanistan's Mathematician "Abu Jafar Muhammad ibn Musa Al-Khwarizmi" in Ninth Century.

(1) An Algorithm is a set of Rules for Carrying out Calculation either by hand or on a machine.

(2) An Algorithm is a Sequence of Computational steps that transform the Input into the output.

(3) An Algorithm is a Sequence of operations performed on data that have to be organized in data structures.

(4) A finite set of Instructions that specify a sequence of operations to be carried out in order to solve a specific problem or class of problems is called an Algo.

(5) An Algorithm is an Abstraction of a program to be executed on a physical Machine.

(6) An Algorithm is a finite set of Instruction that accomplishes a particular task.

"An Algorithm can be defined as a sequence of definite & Effective instructions while terminates with the production of Correct output from the given Input."

→ Algorithm that are definite & effective are also called "Computational procedures".

→ The study of algorithm includes many important and active areas of research.

There are 4 Distinct areas of study

① How to devise an Algorithm.

② How to validate an Algorithm.

③ How to analyze an Algorithm

④ How to test a Program.

Algorithm (Cont…)

In addition every algorithm must satisfy following criteria:

- Input – Zero or more quantities externally supplied
- Output – At least one output is produced
- Definiteness – Each Instruction must be clear & unambiguous
- Finiteness – Algorithm must terminate after finite number of steps
- Effectiveness – Instruction should be easily understandable and sufficiently Simple and basic

Why Study Algorithm?

Why Study Algorithm →

Processor of speed increases ⟶ Performance increases ?

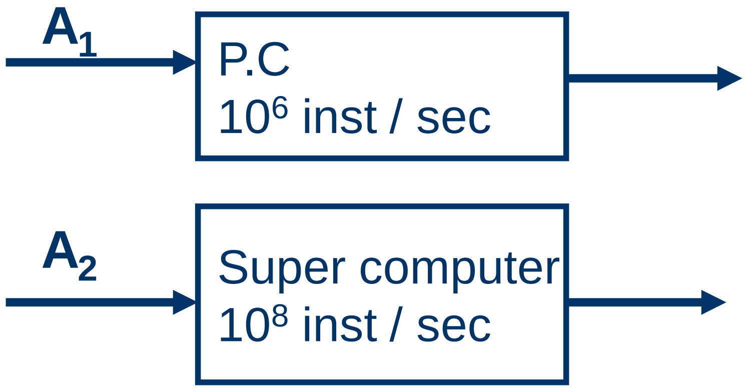
$$\begin{tikzcd}
\text{– Problem size Matters} \arrow[r] & \text{Large size} \arrow[r, "\text{affects}"'] \arrow[d] & \text{Performance} \\
& \text{Longer Computation time} \arrow[d] & \\
& \text{Slower the Results} &
\end{tikzcd}$$

– The Study of Algorithm gives ~~us~~ a language to express performance as a function of Problem Size.

Example

Two algorithms on two systems

- Algorithm A_1 50 n lg n
- Algorithm A_2 2 n^2



For n = 10^6

Time taken by Super Computer

$= 2(10^6)2/ 10^8 = 20,000$ sec

Time taken by P .C.

$= 50 (10^6 \lg 10^6) / 10^6 = 1,000$ sec

Algorithm Specifications

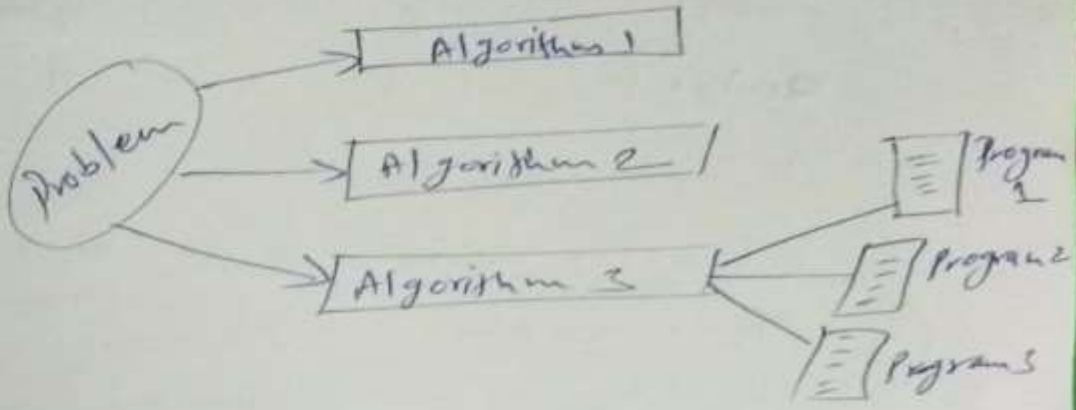
Algorithm Specifications

ALGORITHM vs PROGRAM: The PROGRAM does not have the Finiteness condition.

Algorithm v/s Program →

Programming Structure →

1. Problem ?
2. " Analysis
3. Flowchart
4. Algorithm ✓
5. Pseudocode
6. Coding
7. Testing
8. Documentation.



– In Computational Theory, Algo & program are Different.

Program → Does not have to satisfy the Finiteness Condition
↳ Implementation Phase

Algorithm → have a termination condition.
↳ The designing Phase of a Problem

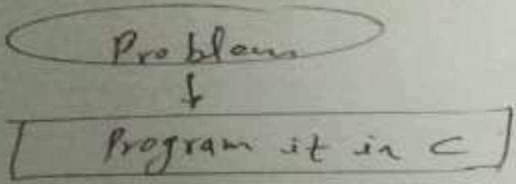
Program → Implementation phase of a designed Algo.
So the Concrete expression of an Algorithm in a particular programming language is called a program.

Pseudo code

- Pseudocode is an English language like representation of the code required for an algorithm.
- It is partly English, partly structured code.
- The English part provides a relaxed syntax that is easy to read.
- The code part consists of an extended version of the basic algorithmic constructs-sequence, selection and iteration.

Analysis of Algorithm

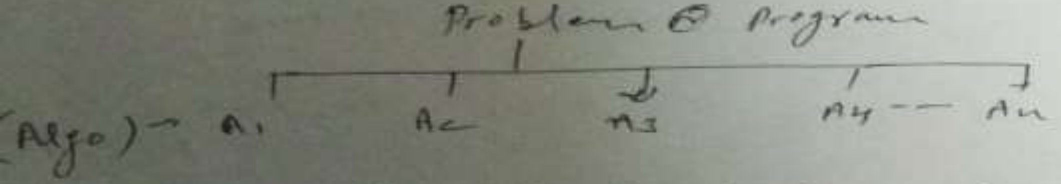
– Solving Problem in Computer Science, Before writing Program, we can write a Informal Description of solution Called Algorithm.



– Algo is in Informal due to It is like to Communicate before Implementation.

For writing a Program, we need Algo.

If you a Problem, You may have many solution.



– Every Algo can be Implemented in form of ~~[illegible]~~ Program.

– we have to need to know which Algo is good in terms of Time & memory (space).

– Time & memory take less considered as good.

– Design ⟶ how can we Design $\overset{\text{various}}{\text{Algo}}$ for given Problem
&
Analysis ⟶ how to Analyze these Algo in respect of Time & space
of Algorithm

Analysis of Algorithm

- Issues:
  - Correctness
  - **Time Efficiency**
  - **Space Efficiency**
  - Optimality

- Approaches:
  - **Theoretical Analysis**
  - **Empirical Analysis**

Time Efficiency

- Time T (*P*) taken by a program *P* is the sum of the Compile time and run (or execution) time.
- Program once compiled can be run several times.
- Compile time does not depend on the instance characteristics.

Space Efficiency

- Space Complexity is the amount of memory an algorithm needs to run to completion.
- Space needed by an algorithm can be sum of following components:
  - ☑ A fixed part that is independent of the characteristics of the input & outputs. This part typically includes the instruction space, space for variables, constants etc.
  - ☑ A variable part consists of the space needed by component variables whose size is dependent on the particular problem instance being solved.

Theoretical Analysis of Time Efficiency

Time efficiency is analyzed by determining the number of repetitions of the *basic operation* as a function of *input size*.

Basic operation: the operation that contributes the most towards the running time of the algorithm.

input size

$$T(n) \approx c_{op}C(n)$$

running time

execution time for basic operation or cost

Number of times basic operation is executed

Empirical/Experimental Analysis of Time Efficiency

- Select a specific (typical) sample of inputs

- Use physical unit of time (e.g., milliseconds)

  or

  Count actual number of basic operation’s executions

- Analyze the empirical data

Algorithm Design Strategies / Techniques

Algorithm Design Strategies / Techniques

- Brute force
- **Divide and conquer**
- Decrease and conquer
- Transform and conquer
- **Greedy approach**
- **Dynamic programming**
- **Backtracking**
- **Branch-and-Bound**
- Space and time tradeoffs