

Design and Analysis of Algorithms

Introduction to Algorithms

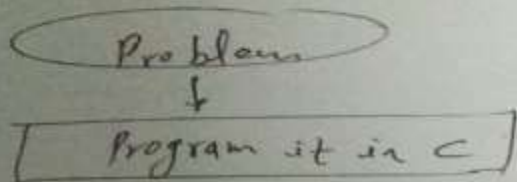
Content

- ▣ Analysis of Algorithms
- ▣ Time Efficiency
- ▣ Space Efficiency
- ▣ Theoretical Analysis of Time Efficiency
- ▣ Empirical/Experimental Analysis of Time Efficiency
- ▣ Algorithms Design Strategies/Techniques

A decorative graphic on the left side of the slide, consisting of a light green vertical bar and a white rounded rectangle with a dark blue horizontal bar extending from its right side.

Analysis of Algorithm

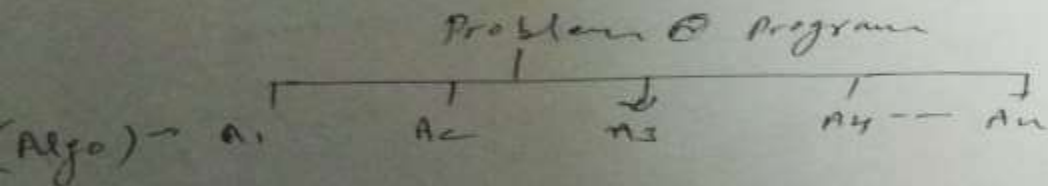
- Solving Problem in Computer Science, Before writing program, we can write a Informal Description of solution called Algorithm.



- 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 solutions.



- Every Algo can be implemented in form of ~~the~~ 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 ^{various} Algo for given problem
- Analysis of Algorithm → how to Analyze these Algo in respect of Time & Space

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.

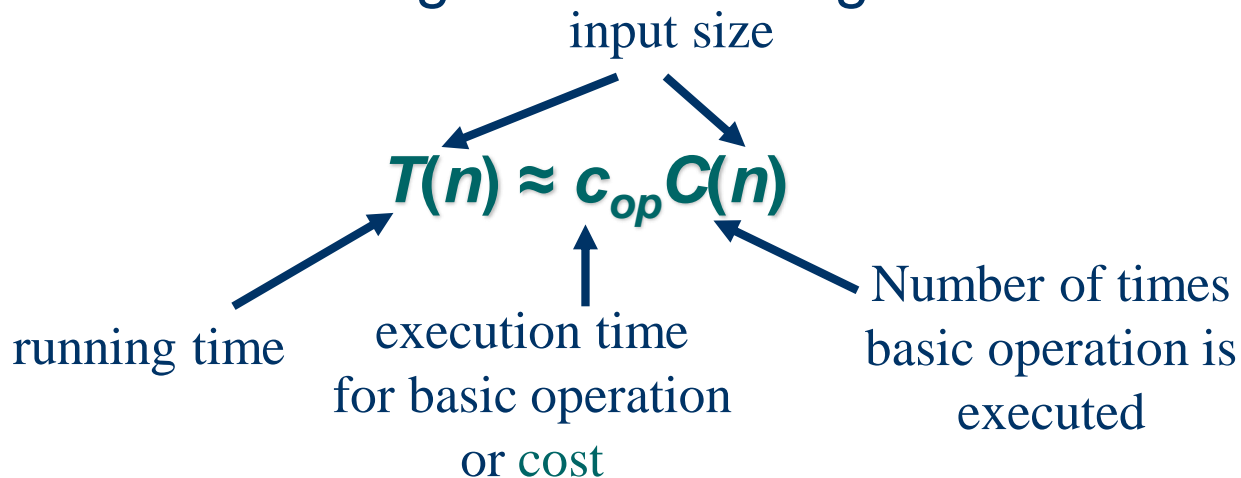
Space Efficiency

- ❏ 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.



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