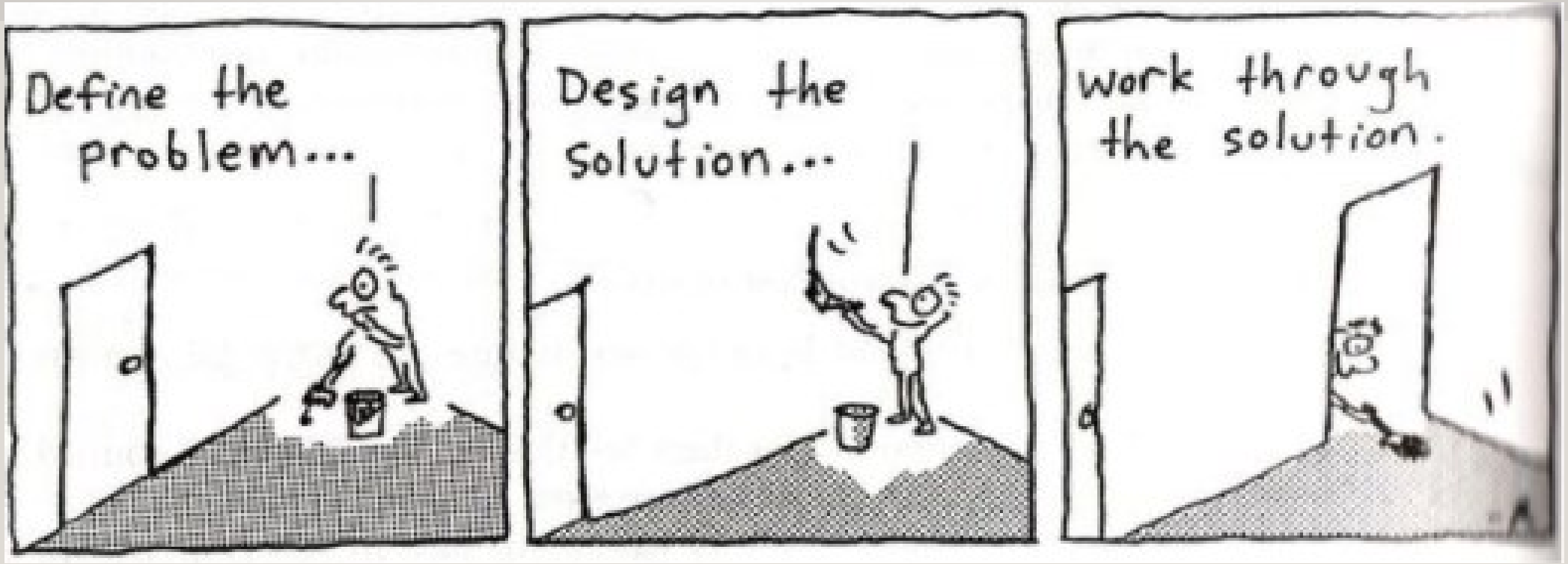
INTRODUCTION TO ALGORITHMS

TEXT BOOK: INTRODUCTION TO ALGORITHMS, THIRD EDITION

(**THOMAS H**. **CORMEN**. CHARLES E. LEISERSON. RONALD L. RIVEST. CLIFFORD STEIN)

- A Program Must be systematically and properly designed before coding begins. This design process results in the construction of an Algorithm.



BASIC CONCEPT OF ALGORITHM

Outcome:

1. Describe elements of an Algorithm.
2. SDLC
3. Represents an algorithm using pseudo code and flowchat

ALGORITHM

- There is no general definition of algorithm accepted.
- But, we can say that the algorithm is :

1. A step by step problem solving procedure
2. A sequence of instruction that tell how to solve particular problem.
3. A set of instruction for solving problem, especially on a computer.
4. A computable set of steps to achieve a desired result.

However, most agree that algorithm has something to do with defined generalized process to get "output" from the given set of "Input".

DEFINITION

- It is a step by step problem solving process in which a solution is arrived in a finite amount of time.

OR

- An algorithm is a finite list of sequential steps specifying actions that if performed result in solution of a specific problem

ALGORITHM AND LOGICAL THINKING

- To develop the algorithm , the programmer needs to ask:
- What data has to be fed into the computer?
- What information do I want to get out from the computer.

- Logic: Planning the processing of the program. It contains the instruction that cause the input data to be turned into the desired output data.

ALGORITHM REPRESENTATION

- There is no standard method of algorithm representation exist. Commonly used representations are:
- Flowchart
- Pseudo code
- Tree Representation

FROM ALGORITHM TO PROGRAM

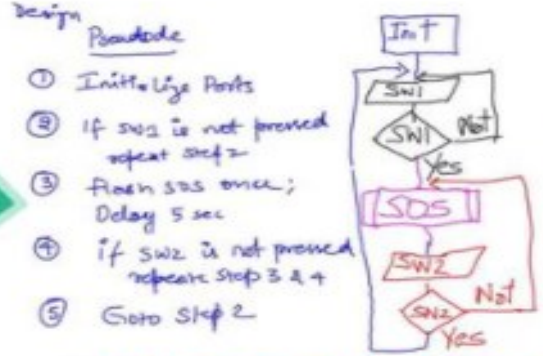
Algorithm:

- Easy to understand
- In plain English language

Program

- Communicate to computer.
- Can be regarded as a "formal expression" of an algorithm.

From Algorithm to Program



ALGORITHM

a sequence of instructions describing how to do a task (or process)

PROGRAM

a set of instructions which are commands to computers to perform specific operations on data

ALGORITHM AND PROGRAM

Algorithm	Program
A step by step problem solving procedure.	A program is a set of instruction which are commands to computers to perform specific operations on data.
A sequence of instruction that tell how to solve a particular problem	Instruction within a program are organized in such a way, when the program is run on a computer; it results in the solution of a problem.
A set of instruction for solving a problem, especially on a computer.	
A computable set of steps to achieve a desired result.	
Written in pseudocode or flowchart.	Written in any programming language.

Program Development Life Cycle

i. **Problem solving phases**

Consist of problem definition and algorithm design

Phase 1: Problem Definition (Analysis)

Phase 2: Algorithm design

Phase 3: Algorithm implementation

Phase 4: Program testing

Phase 5: Program maintenance

ii. **Implementation phases**

Consist of algorithm implementation, program testing and program maintenance phases.

Phase 1: Problem Definition (Analysis)

- The problem is defined to obtain a clear understanding of the problem requirement.
- It is important before come out with the solution algorithm to do the following:
  a. Analyze the problem thoroughly.
  b. Understand the requirement of the problem and required solution.
  c. Divide the problem into sub-problem if the problem is complex.
- The most important and major requirement that need to be specified during problem definition (analysis):
  a. Input
  b. Process
  c. Output
- However, to get a complete problem specification, the following questions should be asked during problem definition (analysis):
  a. What are the input data?
  b. What are the output (desired) data?
  c. What formula is to be used?
  d. What other assumptions or constraints can be made?
  e. What is the expected output screen?

Phase 2: Algorithm design

- The specifications derived earlier in the analysis phase are translated into the algorithm. An algorithm is a step-by-step sequence of precise instructions that must terminate and describes how the data is to be processed to produce the desired outputs. The instruction may be expressed in a human language.
- An algorithm must satisfy some requirements:
  - *Input* : It must have zero or more input
  - *Output* : It must produce at least one output.
  - *Definiteness* : Every step in algorithm must be clear as to what it is supposed to do and how many times it is expected to be executed.
  - *Effectiveness* : It must be correct and efficiently solve the problem for it is designed.
  - *Finiteness* : It must execute its instruction and terminate in a finite time.

Algorithm Representation

- Almost every program involves the steps of **input**, **processing**, and **output**.
- Therefore graphical representation are needed to separate these three steps.
- An algorithm can be written or described or represent using several tools:
  - **Pseudo code**

    Use English-like phrases to describe the processing process. It is not standardized since every programmer has his or her own way of planning the algorithm.
  - **Flowchart**

    Use standardized symbol to show the steps the computer needs to take to accomplish the program's objective. Because flowcharts are cumbersome to revise, they have fallen out of favor by professional programmers. Pseudo code, on the other hand, has gained increasing acceptance.

Pseudo Code Style

Style 1	Style 2	Style 3 (Modular design)
Problem 1. Start 2. Task 3. Action 4. End	Problem 1. Start 2. Subproblem 1 Task 1,1 Action 1,1,1 Action 1,1,2 3. Subproblem 2 Task 1,2 Action 1,2,1 Action 1,2,2 4. End	Problem 1. Start 2. Subproblem 1 3. Subproblem 2 4. End 2. Subproblem 1 Task 1,1 Action 1,1,1 Action 1,1,2 3. Subproblem 2 Task 1,2

Pseudo Code Example

Design an algorithm to find the area of a rectangle

The formulas: area = length * width

Input	Process	Output
Input variable: length width	Processing item: area Formula: area = length x width Step / Solution algorithm: get input calculate area display output	Output: area

```
Start
  Read length
  Read width
  Calculate area of a rectangle
  Display area of a rectangle
End
```

OR

```
Start
  Input length
  Input width
  Calculate area of a rectangle
  Output area of a rectangle
End
```

OR

```
Start
  Prompt the user to enter a length of a rectangle
  Prompt the user to enter a width of a rectangle
  Calculate the area of a rectangle
  Display the area of a rectangle
End
```

WHY ALGORITHM IS IMPORTANT

- Describe the steps needed to perform a computation
- Important for writing effective code.
- Code that execute faster and which uses less memory.

SEQUENCE STRUCTURE

- With sequence of structure, an action or event is performed in order, one after another.
- A sequence can contain any number of events but there is no chance to branch off and skip any of the events.
- Once you start a series of events in sequence, you must continue step by step until the sequence ends.

SEQUENCE EXAMPLE

- Problem
  - Design a software to calculate the sum of two numbers entered by user.

  Steps: Program Specification
  - Understand what the problem is.
  - What is needed to solve it
  - What solution should provide for the problem
  - Input to the problem
  - Expected outcome
  - Any special constraints/condition/formulas to be used

Sequence Example

- Problem
  - Design a software to calculate the sum of two numbers entered by user.

Input	Process	Output
Input variable: num1 num2	Processing item: sum Formula: sum = num1 + num2 Step / Solution algorithm: get input calculate sum display output	Output: sum