Algorithm analysis & design

Introduction to Algorithms

Presented By:

T.A. Asmaa Hamad El-saied

E-mail: eng.asmaa134@gmail.com

Lab 1

FCI, Minia University

Agenda

- Introduction
- Algorithm Design.
- Examples

Introduction

- What’s algorithm…?!
- Why algorithm…?!
- Is It Important?!
- Goal
- Before and After!

What's Algorithm?

- Set of finite steps to solve certain problem

- any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output.

- a Finite set of instructions that, if followed, accomplishes a particular task.

Why Algorithm?

- Save resources
- Save time
- Save money

Is It Worth?

- Real examples…
  - **Fibonacci**: recursive vs. loop vs. Dynamic Pro.

    (N = 30, 40, 50)
  - **Median filter**: quick sort vs. counting sort

    (WinSize = 11 **or** 15)
  - **String similarity:** recursive vs. dynamic prog.

    (S1 ="plynomialgood" S2 = "exponentialbad")

Is It Worth?

- It's Crucial CS Course!
  - 4 CS Crucial Courses (according to IEEE-ACM)
    1. Theory of computation “What can be computed?”
    2. **Algorithms** and data structures “Compute it efficiently”
    3. Programming methodology and languages “Code it! different paradigms”
    4. Computer elements and architecture “understand the destination”

Is It Worth?

- It's Core Interview Question!
  - Ask your graduate colleagues!!

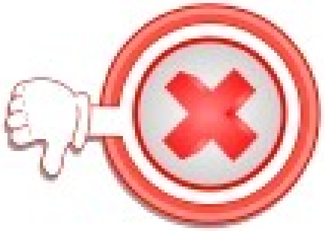
Goal

- Think…
- Design…
- Analyze…

Before & After!

- Before algorithm: Write code to solve problem
- After algorithm: Write **EFFICIENT** code to solve problem

- **It’s a course!!**



- **It’s a skill and attitude**

RESOURCES

- **Textbook:-**

1. Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein

   Introduction to Algorithms. 3rd ed. MIT Press, 2009.

2. Anany Levitin, Introduction to the design and analysis of algorithms

   2nd Edition, 2007.

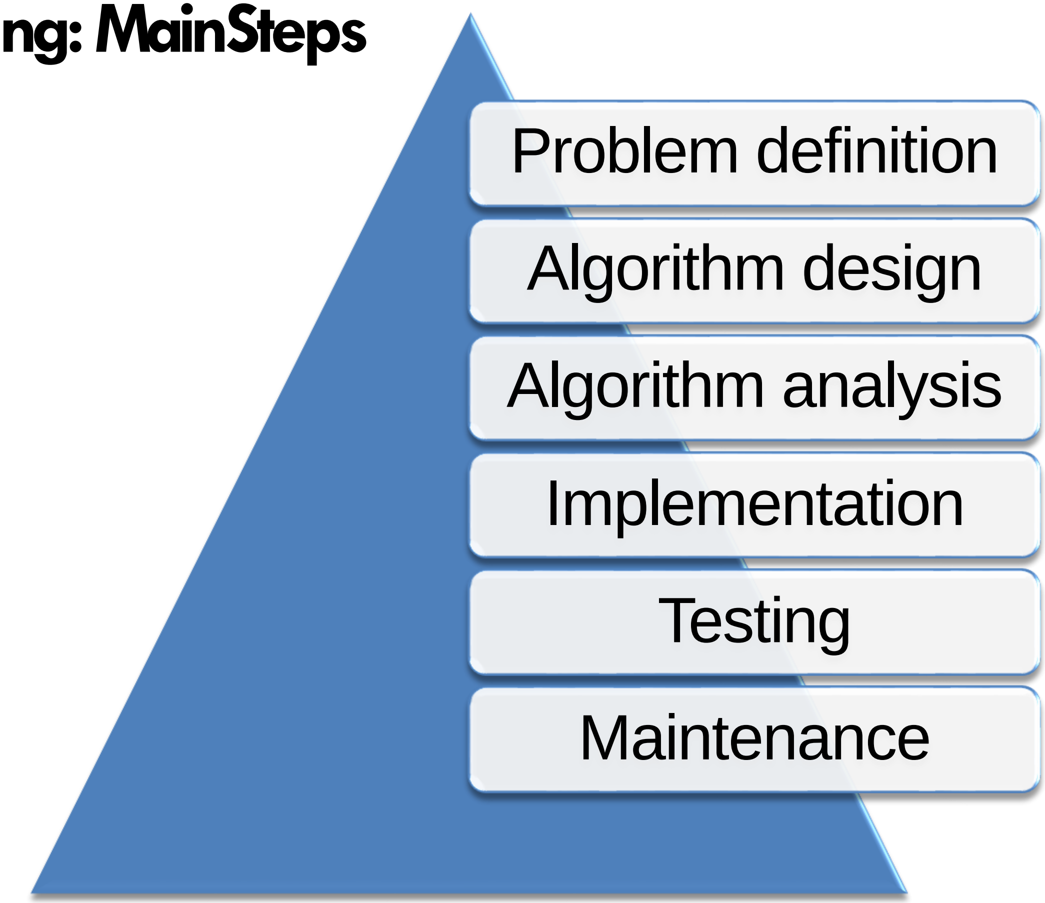
- **Online Courses:-**

1. [Stanford] Algorithms: Design and Analysis: Videos, Join the course

2. [MIT] Introduction to Algorithms: Videos

Problem Solving: Main Steps

- Programming is a process of problem solving
- **Problem Solving: MainSteps**



Problem definition

Algorithm design

Algorithm analysis

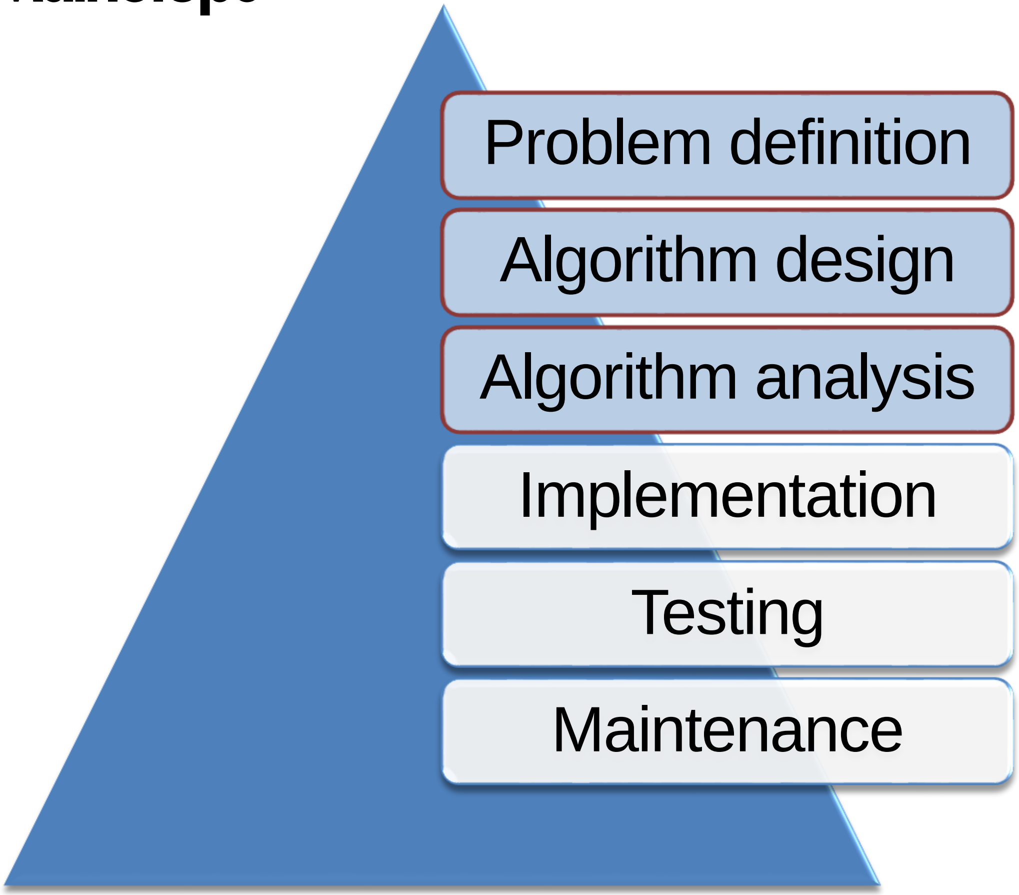
Implementation

Testing

Maintenance

Problem Solving: Main Steps

- **Problem Solving: MainSteps**



Algorithm design

Algorithm design: How to describe Algorithm?

- Algorithm can be described/ represented in three ways.

1. **Natural language like English:**
2. **Graphic representation called flowchart:**
3. **Pseudo-code Method:**

   In this method, algorithms are written in a format that is closely related to high level programming language structures.

- From our Objectives is Design algorithms using Pseudo-code.

Pseudo-code conventions:

1. Comments begin with // and continue until the end of line.
2. Blocks are indicated with matching braces {and}.
3. An identifier begins with a letter. The data types of variables are not explicitly declared.
4. Assignment of values to variables is done using the assignment statement.

   <Variable>:= <expression>; Or <Variable> ← <expression>;

5. There are two Boolean values TRUE and FALSE.
   - Logical Operators AND, OR, NOT
   - Relational Operators <, <=,>,>=, =, !=

Pseudo-code conventions:

6- The following looping statements are employed.

For, while and repeat-until

- While Loop:

```
While < condition > do
    <statement-1>
    <statement-n>
End While
```

Pseudo-code conventions:

- For Loop:

```
For variable: = value-1 to value-2 do
    <statement-1>
    <statement-n>
End For
```

- repeat-until:

```
repeat
<statement-1>
        .
<statement-n>
until<condition>
```

Pseudo-code conventions:

7- **A conditional statement** has the following forms.

```
If <condition> then <statement>
If <condition> then <statement-1>

Else <statement-1>
```

Case statement:

```
Switch (expression)
    case 1 : <statement-1>

    case n : <statement-n>
    default : <statement-n+1>
End switch
```

Pseudo-code conventions:

8- Input and output are done using the instructions

read & write (print).

9- The heading of algorithm takes the form,

Algorithm Name (Parameter lists)

Algorithm Design: Examples

Example 1:

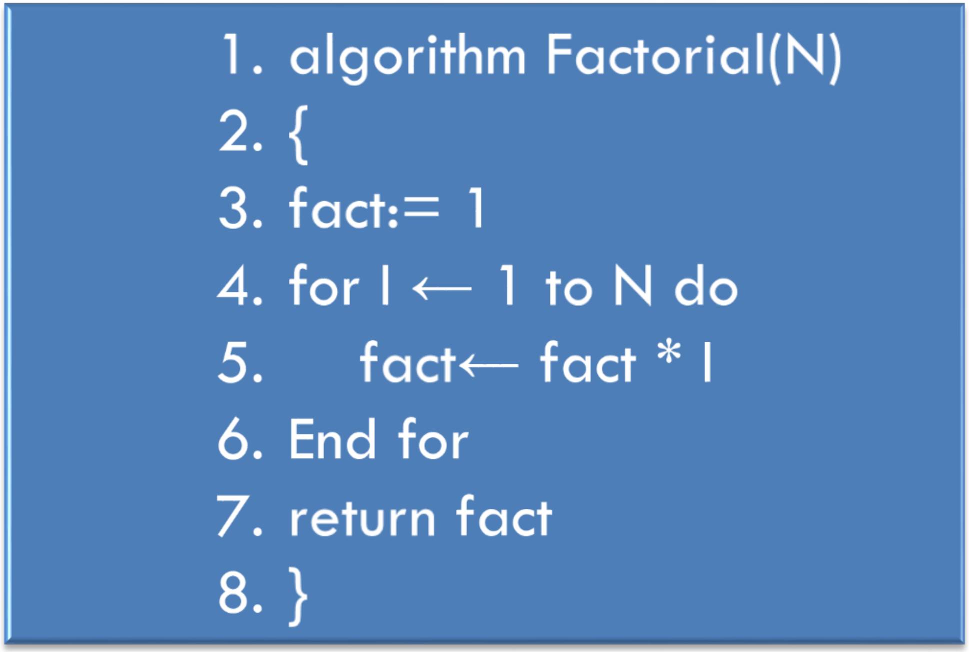
- write a Pseudo Code for finding the maximum number of ‘n’ given numbers in array A.

```
algorithm Max(A,n)
// A is an array of size n
{
Max:= A[1]
for I ← 2 to n do
    if A[I] > Max then
        Max← A[I]
    End if
End for
return Max
}
```

Algorithm Design: Examples

Example 2:

- write a Pseudo Code to calculate the factorial of a number (N).



```
algorithm Factorial(N)
{
fact:= 1
for I ← 1 to N do
    fact← fact * I
End for
return fact
}
```

Algorithm Design: Examples

Example 3:

- write a Pseudo Code with a natural number, N, as its input which calculates the following formula and writes the result in the standard output:

$$S = \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{N}$$

```
algorithm formula(N)
{
K:=2 and S:= 0
While K <=N do
    S← S+ 1/K
    K=K+2
End While
return S
}
```

Thanks