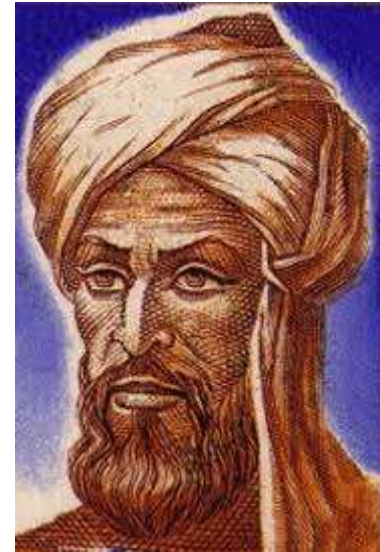


# Advanced Algorithms & Data Structures

# Muhammad ibn Musa al - Khwarizmi Father of Algorithm



## Definition:

An *algorithm* is a finite set of instructions that accomplishes a particular task.

All algorithms must satisfy the following criteria.

## Characteristics:

1. Input : Zero or more quantities are externally supplied.
2. Output : At least one quantity is produced.
3. Definiteness : Each instruction is clear and unambiguous.

Statements such as “add 6 or 7 to x” is not permitted”

4. Finiteness : The algorithm should terminate after a finite number of steps.
5. Effectiveness : Instruction is basic enough to be carried out.

- Algorithms that are effective and definite are called computational procedures.

**Example** : The operating system of a digital computer.

- To achieve the criterion of definiteness algorithm are written in a programming language.
- A program is the expression of an algorithm in a programming language.

## Areas for the study of algorithms:

1. How to devise an algorithm
2. How to validate an algorithm
3. How to analyze an algorithm
4. How to test a program
  - Debugging
  - Profiling

## 1. How to devise an algorithm

To design new algorithm, we apply a existing design strategy.

Some of the existing strategies are

Divide and Conquer

Greedy Method

Dynamic Programming

Back Tracking

Branch and Bound etc.

## 2. How to validate an algorithm

To check whether the algorithm giving correct output or not for all possible inputs.

### 3. How to analyze an algorithm

Based on performance measures such as space and time

- i) How much time required
- ii) How much space required

### 4. How to test a program

- Debugging
- Profiling

Debugging is a process of executing program on sample data sets, to determine whether any fault results occur. If so correct them.

Profiling is a process of executing a correct program on sample data sets and determine how much time/space required by program

## Representation and Structure of an Algorithm:

We can describe an algorithm in many ways.

- ✓ We can use a natural language like English, but if we select this option, we must make sure that the resulting instructions are definite.
- ✓ Graphic representations called flow charts are another possibility, but they work well only if the algorithm is small and simple.
- ✓ We can also represent algorithms using pseudo code that resemble C and Pascal.



We present most of our algorithms using pseudo code that resembles C and Pascal.

1. Comments begin with // and continue until end of the line.
2. Blocks are indicated with matching braces: { and }.
  - i. A compound statement
  - ii. Body of a procedure.
3.
  - i. An identifier begins with a letter.
  - ii. The data types of variables are not explicitly declared.
  - iii. The types will be clear from the context.
  - iv. Whether a variable is global or local to a procedure will also be evident from the context.
  - v. We assume simple data types such as integer, float, char, boolean, and so on.

4. Assignment of values to variables is done using the assignment statement.

$\langle \text{variable} \rangle := \langle \text{expression} \rangle$

5. There are two boolean values **true** and **false**. To produce these values, logical operators **and**, **or** and **not** and the relational operators  $\langle, \leq, =, \neq, \geq$  and  $\rangle$  are provided.

6. Elements of multidimensional arrays are accessed using  $[$  and  $]$ . For example the  $(i,j)$ th element of the array  $A$  is denoted as  $A[i,j]$ .

7. The following looping statements are used: while, for and repeat until.

i) The general form of a **while** loop:

```
while( condition ) do  
    {  
        statement_1;  
        :  
        statement_n;  
    }
```

ii) The general form of a **for** loop:

```
for variable := value1 to value2 step step do
```

iii) The general form of a repeat-until loop:

repeat

<statement 1>

:

<statement n>

until ( condition )

The statements are executed as long as condition is false

8. A conditional statement has the following forms:

if < condition > then < statement >

if < condition > then

< statement 1>

else

< statement 2>

9. Input and output are done using the instructions read and write.

Ex: read n;

write n;

## 10. Procedure or function starts with the word **Algorithm.**

General form :

```
Algorithm Name( <parameter list> )  
{  
    body  
}
```

where *Name* is the name of the procedure.

Simple variables to procedures are passed by value.

Arrays and records are passed by reference

Write an algorithm to find sum of n numbers in the given list

Algorithm sum(a,n)

{

s:=0;

for i:=1 to n do

s:=s+a[i];

return s;

}

# Write an algorithm to find factorial of a given number.

# SAMPLE QUESTIONS

- Explain the Importance of Algorithm to Solve Real World Problems
- Summarize the Characteristics of Algorithm
- What is the difference between an algorithm and a program
- Describe the concept of recursion in algorithms. Provide an example algorithm that uses recursion
- What is an algorithm? Provide a brief definition and explain its importance in computer science.
- What is effectiveness in algorithms
- What is finiteness in algorithms
- In how many ways can we describe an algorithm
- What are the steps included in writing an algorithm
- How do we validate an algorithm
- How to Analyse an algorithm



