Data Structure and Algorithm

What is Algorithm?

A sequence of unambiguous instructions for solving a problem.

Can be considered as a procedural solution to a problem.

A finite and definite procedure for solving a problem.

A set of rules for carrying out calculation either by hand or on a machine.

Requirements for a Good Software:

* Clean Design
* Easy maintenance
* Reliable (no core dumps)
* Easy to use
* Fast algorithms
* Efficient data structures
* Efficient algorithms

Classification of Data Structure

History of Algorithm:

Comes from “algorism” which in turn comes from the name of a person textbook

Author, Abu Jafar Muhammed Ibn Musa Al-khwarizmi (c.825 A.D.)

The last part of his name (literally “native of [the town] Khwarizmi”) was distorted into “algorism,” which came to mean “the art of calculation” – now called arithmetic.

>PRIMITIVE

* Integer
* Floating point
* Character
* Pointer

>NON PRIMITIVE

* Array
* List

Euclid’s Algorithm.

Most famous algorithm in history dates well before the time of the ancient Greeks.

Calculating the greatest common divisor of two integers.

By 1950, the word “algorithm” was associated with Euclid’s Algorithm (c. 300 B.C.) which first appeared in Elements, book vii, propositions I and ii.

* File
* - LINEAR LIST

-Stack

- Queue

* - NON LINEAR LIST
* Graph
* Tree

Expressing an Algorithm

English/Human language description

Pseudocode

High level programming language

Characteristics of Algorithm:

Input

Output

Definiteness

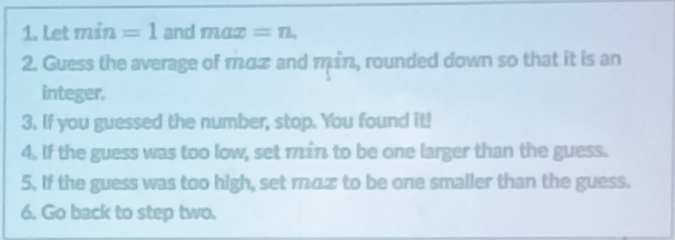
Effectiveness

Termination

English/Human Language:

* Easily understandable by humans.
* Often ambiguous and hard to understand by a machine.

Example:



Pseudocode:

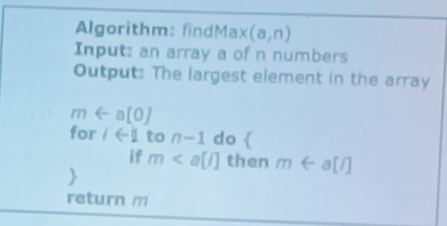
A way to describe algorithms that does not depend on a particular programming language, but that is unambiguous.

Easily implanted, and easy to read for people.

Often the first pass at a solution for a problem.

To be followed by an implementation in an actual programming language.

Example:

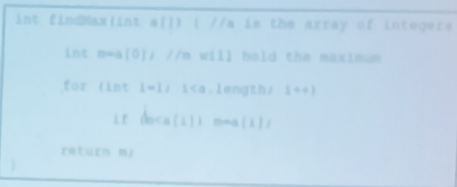


Programming Language:

Write programs in some kind of “high-level” programming language that is then translated by another program, called the compiler or interpreter, into binary language.

E.g Java, C, C++, Lisp, Fortran, Perl, Python, ML.

Example:



Notation of Algorithm:

Space Complexity:

The amount of memory required by an algorithm to run to completion.

Some algorithms may be more efficient if data completely loaded into memory.

Fixed part: The size required to store certain data/variables, that is independent of the size of the problem.

Variable part: Space needed by variables, whose size is dependent on the size of the problem.

Problem -> Algorithm

Input -> Computer -> Output

Problem Solving Main Steps:

Problem Definition

Algorithm design or specification

Algorithm analysis

Implementation

Time Complexity:

Often more important than space complexity.

Algorithms running time is an important issue.

Testing

Maintenance