

Vistula, IT Faculty, 2014

# Algorithms and Complexity

Dmitry A. Zaitsev

<http://daze.ho.ua>

## Lecture 1:

# An introduction to algorithms and complexity

# Intuitive concept of an algorithm

Abū ‘Abdallāh Muḥammad ibn Mūsā al-Khwārizmī

A precise step-by-step specification of a computations process where the sequence of steps is unambiguously prescribed and at each step it is defined how to obtain a result from the source data.

A classical example – algorithm of Euclid  
Given a two natural numbers  $a$  and  $b$ ,  
find their greatest common divisor (gcd)

Step 1. If  $a=b$  then Stop.  $\text{gcd}=a$

Step 2. If  $a>b$  then  $a=a-b$

else  $b=b-a$

Go to Step 1.

**Given 150 and 275**

Let  $a=150$ ,  $b=275$

Step 1.  $275 \neq 150$

Step 2.  $275 < 150$

$b=275-150=125$

Step 1.  $150 \neq 125$

Step 2.  $150 > 125$

$a=150-125=25$

Step 1.  $25 \neq 125$

Step 2.  $25 < 125$

$b=125-25=100$

Step 1.  $25 \neq 100$

Step 2.  $25 < 100$

$b=100-25=75$

Step 1.  $25 \neq 75$

Step 2.  $25 < 75$

$b=75-25=50$

Step 1.  $25 \neq 50$

Step 2.  $25 < 50$

$b=50-25=25$

Step 1.  $25=25$ . **gcd=25**

## Features of algorithm

- Generality – intended for solving a series of similar individual tasks ( $x+y=?$  but not  $2+2=?$ )
- Determinism (unambiguity) – after each step we precisely know which step to execute next
- Constructivity – it stops after a finite number of steps – potential realizability
- Elementarism of steps – each step is so simple that can be implemented unambiguously precise

## Hilbert list and necessity of formalization

- Earning for an universal solver of problems
- List of Hilbert, 1900, 23 problems
- An example of 10<sup>th</sup> problem – solvability of algebraic Diophantine equations:

$$6x^3yz^2 + 3xy^2 - x^3 - 10 = 0 \text{ at } x = 5, y = 3, z = 0$$

- It seems there are problems for which solving there is no algorithm
- To prove it we should work with some formal definition of an algorithm

## Known models of computations

- Turing machine
- Post machine
- Partially recursive function
- Normal algorithm of Markov
- Counter automaton
- Hypothetic machines RAM, RASP
- Cellular automaton
- etc

Thesis of Church - each algorithm can be represented in the form of corresponding Turing machine

## Undecidable, intractable, and tractable

- Undecidable problems – it is proven that an algorithm of their solving does not exist
- Intractable problems – there are algorithms of their solving but it is proven that all the known algorithms require too much time
- Tractable problems – there are algorithms of their solving in a reasonable time

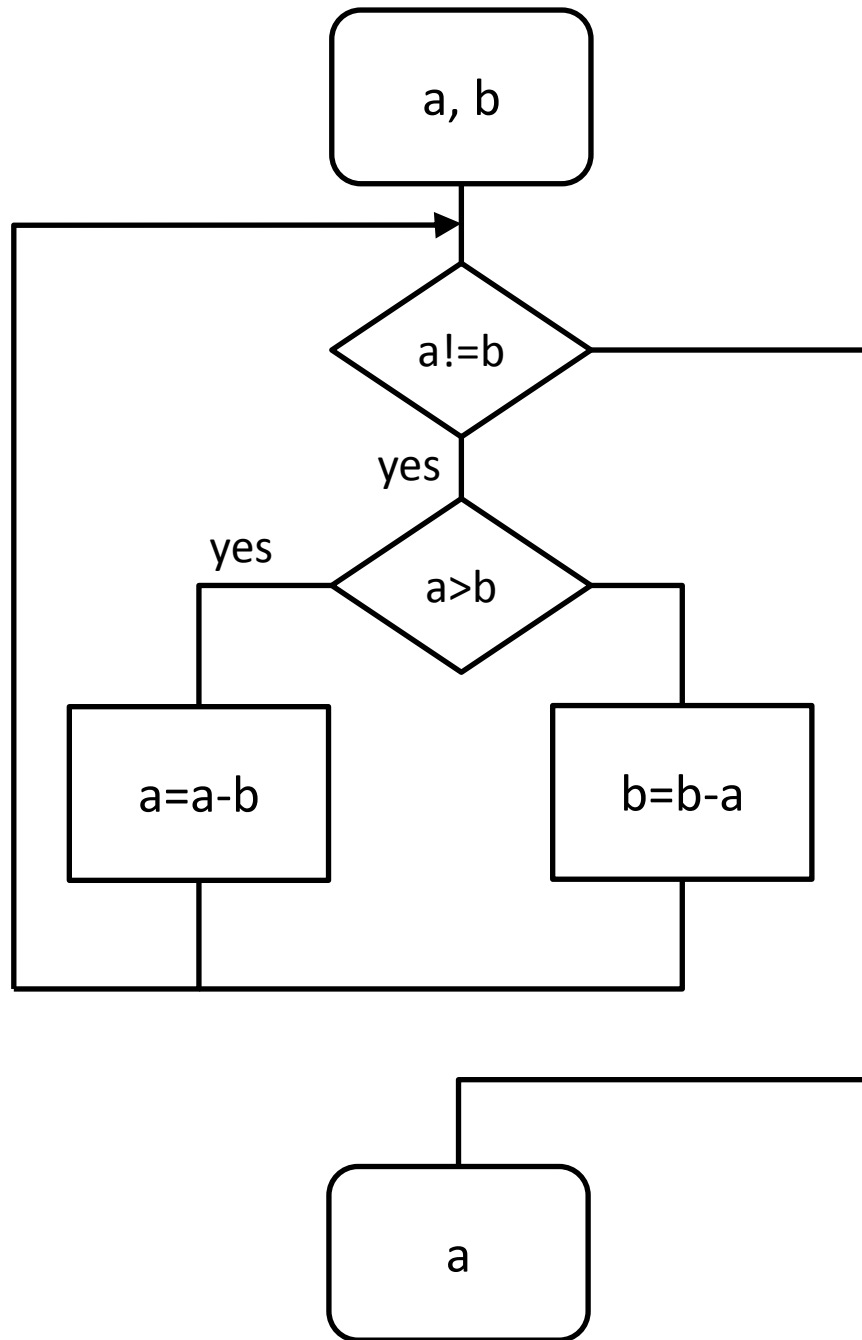
## Correctness of algorithms

- Testing – execute the algorithm on a given input data and compare the result with expected one
- Correctness proof – prove formally that algorithm produces a correct result for any given input data

## Algorithms and programs

- Step-by-step verbal description
- Pseudo-language description (PDL)
- Flow chart
- A program in an algorithmic language
- A program in some computer language – a sequence of instructions of a certain processor





**gcd: Flow chart and  
program in C language**

```
int gcd( int a,int b )
{
    while( a != b )
        if( a > b )
            a = a - b;
        else
            b = b - a;
    return( a );
}
```