

Algorithmization and Programming of Solutions

Part I

TOPIC #1

AN INTRODUCTION TO ALGORITHMIZATION

RTU LECTURER OLGA YAKOVLEVA, MG.SC.ING

Table of contents

1. An introduction to algorithmization

- a process of solving tasks using a computer
- the concept of algorithmization and of an algorithms
- the properties of an algorithm

1. An introduction to algorithmization

A PROCESS OF SOLVING TASKS USING A COMPUTER, THE CONCEPT OF ALGORITHMIZATION AND OF AN ALGORITHM, THE PROPERTIES OF AN ALGORITHM

A process of solving tasks using a computer

1. Identify a problem and define a task
2. Understand a task
3. Design a solution
4. Implement the solution
5. Test the solution
6. If there are errors or shortcomings, go back to the step #3, i.e., check the design -> check the implementation -> find and fix the errors
7. Execute the solution

The concept of algorithmization

Algorithmization is a process of development of an algorithm for solving a defined task

Algorithmization is only a part of a process of solving tasks using a computer, because it starts at the step #2 and ends up with the step #3

During the process of an algorithm development an answer to the question: “**How** to solve a defined task?” is found

The result of algorithmization is **an algorithm** for solving a defined task

A process of algorithmization

Before searching for or developing an algorithm it is necessary to clarify, **whether it is possible to solve a defined task**, because some tasks / problems do not have a solution:

- errors / shortcomings of a task definition:
 $a = 5$, $c = a + b$, what is the value of c ?
- Millennium Prize Problems,
https://en.wikipedia.org/wiki/Millennium_Prize_Problems)

During a process of searching for or developing an algorithm it is needed **to analyse theoretical background** of a task, e.g., find a mathematical formula or a method, define input and output data types, etc.

The concept of an algorithm (1/2)

An **algorithm** is:

- a rule or procedure for solving a problem (*Microsoft Terminology 2018. Entry from the Microsoft Language Portal*)
- is a step by step method of solving a problem (<https://www.techopedia.com>)
- a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer (<https://www.dictionary.com>)

The concept of an algorithm (2/2)

As an easy definition of an algorithm, we may say that it is a description of actions it is necessary to perform to achieve a goal or a **fixed sequence of actions**, which means:

- an algorithm consists of actions
- actions are related to each other

The properties of an algorithm

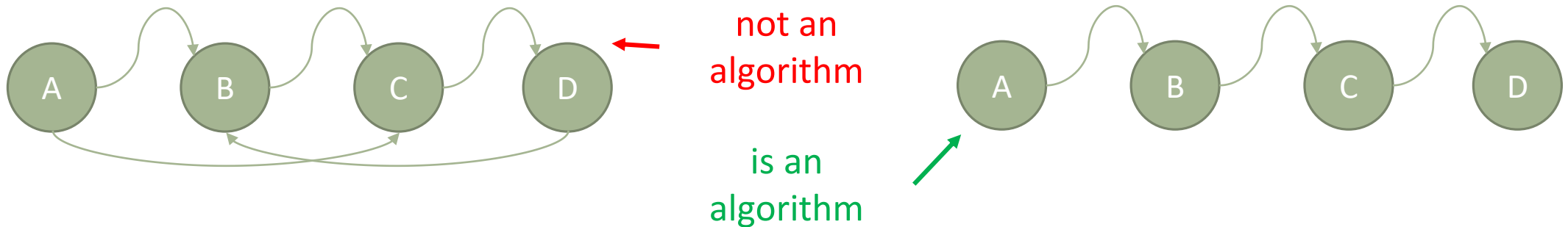
An algorithm have to possess the following properties:

- **discreteness**
- **definiteness** (form the word “definite” or “precise”)
- **finiteness** (from the word “finish”)
- **universality** (in a sense of multi-purpose)

Discreteness

An algorithm have to be split into **separate actions**

It's impossible to proceed to the next action while the previous one is not finished



Definiteness

Each action of an algorithm must have **one and only one interpretation**

E.g., “salt to taste and cook to readiness”

or

“add 1 teaspoon of salt and cook 30 minutes”



not an
algorithm



is an
algorithm

Finiteness

An algorithm have to **guarantee achievement of a goal**

E.g., $a = 5$

$b = \text{user defined value}$

$c = a / b$

or

$a = 5$

$b = \text{user defined value}$

if b is a number and b is not 0, then $c = a / b$



is not an
algorithm



is an
algorithm

Universality

The algorithm should be universal, meaning suitable for tasks with different input values

It is recommended not to bind an algorithm with an implementation

The more universal is an algorithm, the more valuable it is

E.g., a cooking recipe is not a universal algorithm,
but column addition is

$$\begin{array}{r} 1 \\ 12 \\ +19 \\ \hline 31 \end{array}$$

Traditional