

Algorithms, Data Structures and Programming

Introduction

Week 1 | Information systems, 2 course

Lecture Objectives

After this lecture, students will:

Understanding Algorithms

What an algorithm is and its main properties.

Algorithm Notation

Know different ways of writing algorithms, including flowcharts.

Connection to Programs

Understand the connection between algorithms and programs.

Number Systems

Know the main number systems in programming.

What is an algorithm?

Definition

An **algorithm** is a finite sequence of well-defined instructions for solving a specific problem.

History of the term

The word "algorithm" comes from the name of the Persian mathematician **Muhammad al-Khwarizmi** (783-850 AD).



Algorithms in Everyday Life

We use algorithms every day:



Cooking Recipe

A sequence of steps to prepare a finished dish.



Furniture Assembly Instructions

Step-by-step guide.



Route to University

A sequence of actions to achieve a goal.

Example: Tea Making Algorithm

Pour water, turn on kettle, wait for boiling, put in tea bag, pour boiling water, wait, add sugar, stir.

Properties of an Algorithm

For a sequence of actions to be called an algorithm, it must possess the following properties:

Discreteness

Consists of distinct, clearly defined steps.

Determinism

Each step is precisely and unambiguously defined.

Finiteness

Completes in a finite number of steps.

Generality

Applicable to various sets of initial data.

Effectiveness

Leads to obtaining a result.

Methods for Recording Algorithms

Verbal Description

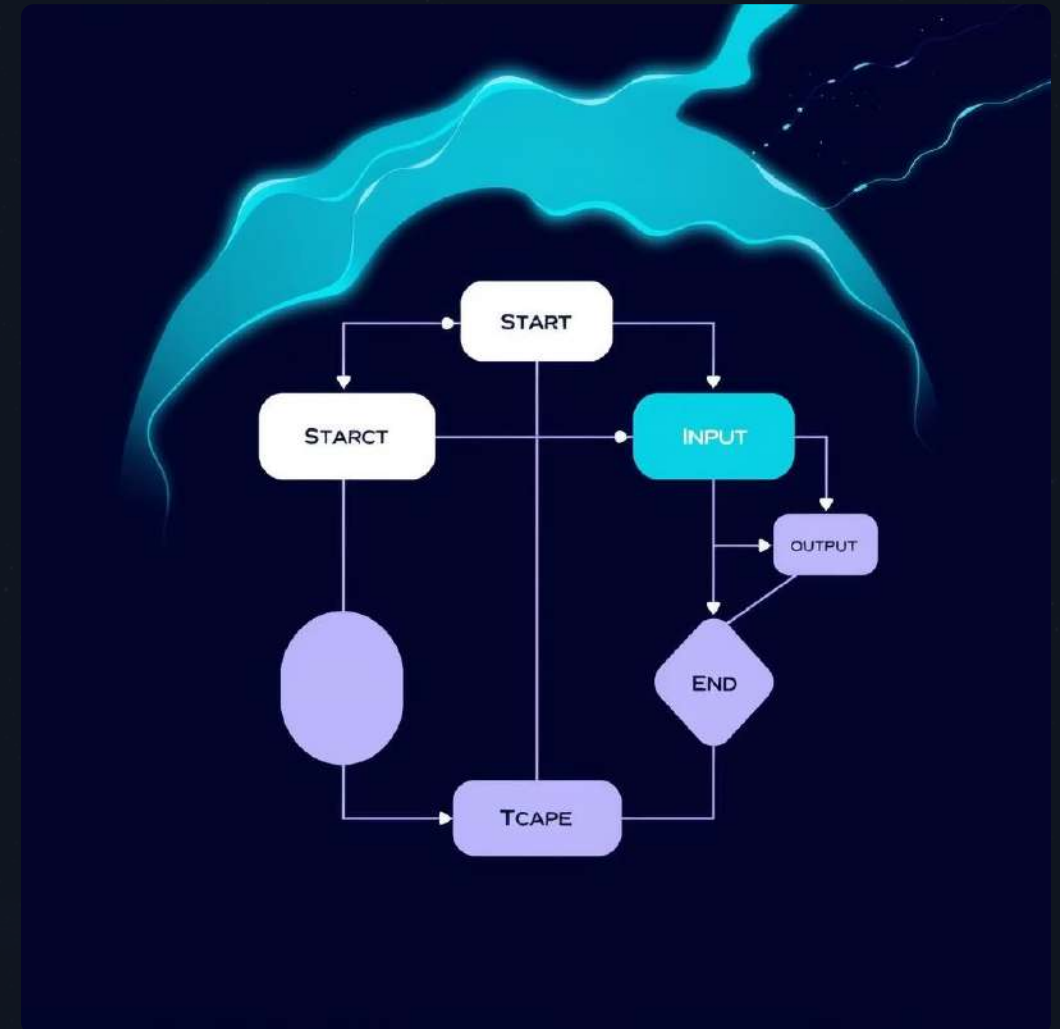
In natural language. Understandable, but can be ambiguous.

Pseudocode

A formalized description using elements of a programming language and natural language. More formal, easily translated into code.

Flowcharts

Graphical representation using geometric shapes. Visual clarity, easy to trace logic.



Flowcharts: Key Components

Flowcharts are a graphical way of representing algorithms, where each step or operation is depicted using symbols of various geometric shapes. This allows for a clear visualization of the program's execution logic.



Oval

Used to denote the **start** and **end** of an algorithm.



Rectangle

Represents the execution of an **action** or **process**.



Diamond

Used to denote a **condition** or a point of **decision** (e.g., "Yes" or "No").



Parallelogram

Indicates data **input** or **output** operations.



Arrows

Indicate the **direction of the sequence of execution** of algorithm steps, connecting blocks.

When constructing flowcharts, it is important to follow rules: there should always be one start and one or more ends. Arrows must always indicate the direction of the control flow. Branches from diamonds should be labeled, indicating the condition for choosing a path.

Program Code

Recording an algorithm in a specific programming language.

```
#include <iostream>
using namespace std;
int main() {
    int a, b, max;
    cin >> a >> b;
    if (a > b) {
        max = a;
    } else {
        max = b;
    }
    cout << "Максимум: " << max << endl;
    return 0;
}
```

Advantages: Can be executed by a computer. **Disadvantages:** Tied to a specific programming language.

From Algorithm to Program

01

Problem Statement

Clear definition of what is given and what needs to be found.

02

Algorithm Development

Creating a sequence of steps for the solution.

03

Program Writing

Translating the algorithm into a programming language.

04

Debugging and Testing

Verifying the correctness of the program's operation.

A program is an algorithm written in a programming language that can be executed by a computer.

Number Systems

A computer only works with binary numbers (0 and 1).

Binary	2	0,1	Internal data representation
Octal	8	0-7	File permissions in Unix/Linux
Decimal	10	0-9	Familiar to humans
Hexadecimal	16	0-9, A-F	Memory addresses, colors

Examples: Color in HTML: #FF0000; Memory address: 0x7FFF5694.

Key Takeaways and Homework

- An algorithm is a clear sequence of actions to solve a problem.
- Five properties of an algorithm: discreteness, definiteness, finiteness, generality, effectiveness.
- Methods of notation: verbal, pseudocode, flowcharts, program code.
- A program is an algorithm written in a programming language.
- Number systems are the basis for data representation in a computer.

Assignment for the next lesson:

- Install the development environment (Visual Studio Code).
- Create a flowchart for the algorithm to determine a leap year.
- Write an algorithm for the problem: "Find the sum of all even numbers from 1 to N".