

Working definition:

An *algorithm* is a system of feasible (well-defined) instructions for (aimed at) solving some problem.

A more general definition:

An *algorithm* is a finite constructive description of a process for (aimed at) solving some problem.

An even more general definition:

An *algorithm* is a finite constructive structure that defines a process for (aimed at) solving some problem.

Constructive means that using this description (structure), an automaton (computer) can perform actions prescribed by the algorithm.

The description, e.g., a system of instructions, is a representation of an algorithm.

Why “for solving” and not “of solving”?

Because in a general case, we cannot know if the algorithm actually solves the assigned problem.

Complete algorithm always solves its problem.

Algorithms existed and were used before any computers were built and computer science created

- Directions for going from one location to another.
- Recipes for cooking.
- Arithmetical algorithms (multiplication, division, etc.).

Algorithms can be represented in many ways:

Physical representation, for example, on a circuit board or silicon chip

Mathematical representation, for example, finite state automata or formal grammars

Linguistic representation, for example, in a programming language (C⁺ , Java, Cobol, etc) or in a natural language (in a recipe book)