

Theory: Computer algorithms

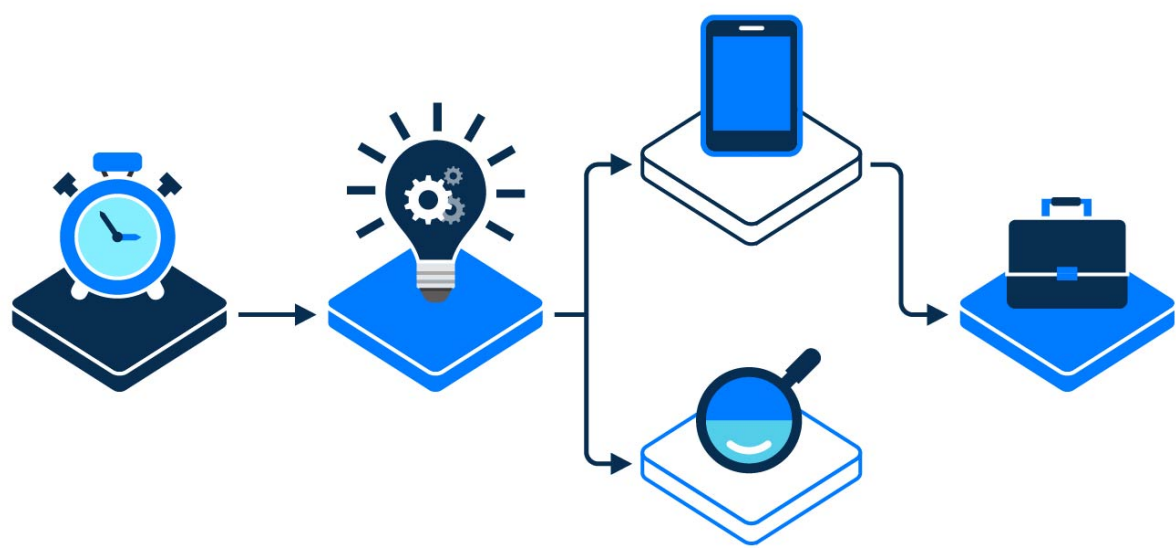
🕒 5 minutes 0 / 4 problems solved

Start practicing

12639 users solved this topic. Latest completion was about 1 hour ago.

§1. Everyday algorithms

You have probably heard something about **algorithms** in real life. Simply put, it is a step-by-step sequence of actions you need to perform to achieve a useful result. It can be an algorithm for cooking a sandwich described by a recipe or an algorithm for getting dressed according to today's weather and your mood.



Among all algorithms, there is one special group called **computer algorithms**. These are ones that are usually created for and utilized by computers. In this topic, we will discuss in detail what computer algorithms are and will explain why it is important to learn them.

§2. Computer algorithms

Computer algorithms are everywhere around us. Your smartphone may guide you through a city from one point to another using a certain algorithm. Other algorithms can control the behavior of your enemies in a computer game. Services like Google or Yahoo apply sophisticated algorithms to provide you with the most relevant results when you use them to search for information on the Web. Algorithms are also used to calculate the trajectory of rockets. And they even help doctors to determine diagnoses correctly!



An important difference between real-life and computer algorithms is that a computer cannot guess what we want to do. If something goes wrong or an algorithm is not clear, a human can adjust the algorithm based on their experience. Computers cannot do the same. Thus, a computer algorithm must be described precisely and unambiguously.

§3. Programs and algorithms

Current topic:

✓ [Computer algorithms](#) ...

Topic is required for:

✓ [Recursion basics](#) ...

✓ [Introducing the first algorithm](#) ...

[Algorithms in Java](#) ...

[Intro to Machine Learning](#) ...

✓ [Algorithms in Python](#) ...

Table of contents:

[1 Computer algorithms](#)

[§1. Everyday algorithms](#)

[§2. Computer algorithms](#)

[§3. Programs and algorithms](#)

[§4. Summary](#)

[Feedback & Comments](#)

As you may know, a program is a sequence of instructions to perform some tasks on a computer. The difference between programs and algorithms is that programs are written using a specific programming language while algorithms are usually described at a higher level than programming language statements. In other words, an algorithm is like an abstract schema, and a program can be its implementation.

All this also means that algorithms are language-agnostic: one algorithm can be implemented using different programming languages. For example, you may use Java, Python, Kotlin, or other languages to implement the same algorithm.

Programming languages usually contain implementations of some basic algorithms for solving typical problems. These algorithms are provided in standard libraries, and software developers can reuse them instead of implementing a new one each time. However, to be able to use such algorithms correctly and efficiently and understand how other developers use them, it is essential to learn these basic algorithms and get familiar with how they work under the hood.

Algorithms from standard libraries cannot cover all possible problems developers can encounter. Thus, sometimes you will need to implement a solution for a problem yourself from scratch. This is another reason why it is essential to learn algorithms: you need to know which one and when to apply and how to implement it efficiently.

§4. Summary


An algorithm is a sequence of actions you need to perform to achieve a useful result. An important group of algorithms is computer algorithms: the ones created for and utilized by computers. There are several reasons why it is essential to learn computer algorithms:

- software developers often encounter tasks of the same type while working on different projects. For such typical tasks, programming languages provide ready-to-use algorithms in standard libraries. To utilize these algorithms efficiently, you need to understand how they work under the hood.
- sometimes you may encounter a problem that is impossible to solve using algorithms from standard libraries. In such cases, you need to implement an algorithm yourself. To be able to do that, you need to know basic algorithmic approaches, their pros and cons, and which one to apply in a particular case.
- often you need not only to write the code yourself but also to read the code written by other developers. If you want to understand the algorithms they might likely use, you need to know basic algorithms and algorithmic approaches as well.
- implementing algorithms might help you to improve your programming skills.

We believe that there are other reasons why learning algorithms is worth it. If you have some, don't hesitate to write them in the comments :)

Good luck with learning algorithms!

 Report a typo

 Thanks for your feedback!

Start practicing

[Comments \(38\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)