

The Importance of Algorithms

Manjunath. R

#16/1, 8th Main Road, Shivanagar, Rajajinagar, Bangalore560010, Karnataka, India

*Corresponding Author Email: manjunath5496@gmail.com

*Website: <http://www.myw3schools.com/>

Abstract: A Computer Program can be viewed as an elaborate algorithm and algorithms are very important in Computer Science for solving a problem -- based on conducting a sequence of specified actions. The best chosen algorithm usually means a small procedure that solves a recurrent problem and makes sure computer will do the given task at best possible manner. In cases where efficiency matters -- a proper algorithm is really vital to be used. An algorithm is important in optimizing a computer program according to the available resources -- often play a very significant part in the structure of artificial intelligence, where simple algorithms are used in simple applications, while more complex ones help frame strong artificial intelligence.

[Manjunath. R. **The Importance of Algorithms.** *N Y Sci J* 2019;12(6):63-65]. ISSN 1554-0200 (print); ISSN 2375-723X (online). <http://www.sciencepub.net/newyork>. 8. doi:[10.7537/marsnys120619.08](https://doi.org/10.7537/marsnys120619.08).

Keywords: Algorithms; Computer; Programming; Instructions; Efficiency; Applications.

Introduction

You might have an algorithm for getting from office to home, for making a chunk of code that calculates the terms of the Fibonacci sequence, or for finding what you're looking for in a retail store. Algorithms are the building blocks of computer programs or sequence of unambiguous instructions (the term 'unambiguous' indicates that there is no room for subjective interpretation) that tells how the problem could be addressed and solved -- which is definitely overblown in their importance like road maps for accomplishing a given, well-defined automated reasoning task -- which always have a clear stopping point.

Long division and column addition are examples that everyone is familiar with -- even a simple function for adding two numbers is implementation of a particular algorithm. Online grammar checking uses algorithms. Financial computations use algorithms. Robotic field uses algorithms for controlling their robot using algorithms. An **encryption algorithm** transforms data according to specified actions to protect it. A search engine like Google uses search engine algorithms (such as, takes search strings of keywords as input, searches its associated database for relevant web pages, and returns results). In fact, it is difficult to think of a task performed by your computer that does not use computer rules that are a lot like a recipes (called algorithms).

The use of computer algorithms (step-by-step techniques used for Problem-solving) plays an essential role in space search programs. Scientists have to use enormous calculations, and they are managed by high-end supercomputers, which are enriched with detailed sets of instructions that computers follow to arrive at an answer. Algorithms have applications in

many different disciplines from **science** to **math** to **physics** and, of course, computing -- and provide us the most ideal option of accomplishing a task. Here is some importance of algorithms in computer programming.

- **To improve the effectiveness of a computer program:** An algorithm (procedure or formula for solving a problem, based on conducting a sequence of specified actions) can be used to improve the speed at which a program executes a problem and has the potential of reducing the time that a program takes to solve a problem.

- **Proper usage of resources:** The right selection of an algorithm will ensure that a program consumes the least amount of memory. Apart from memory, the algorithm can determine the amount of processing power that is needed by a program.

The algorithm for a child's morning routine could be the following:

- Step 1:** Wake up and turn off alarm
- Step 2:** Get dressed
- Step 3:** Brush teeth
- Step 4:** Eat breakfast
- Step 5:** Go to school

The algorithm to add two numbers entered by user would look something like this:

- Step 1:** Start
- Step 2:** Declare variables num1, num2 and sum
- Step 3:** Read values num1 and num2
- Step 4:** Add num1 and num2 and assign the result to sum
- sum ← num1 + num2**
- Step 5:** Display sum
- Step 6:** Stop

Two of these algorithms accomplish exactly the same goal, but each algorithm does it in completely

different way to achieve the required output or to accomplish our task. In computer programming, there are often many different ways – algorithms (any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values as output) -- to accomplish any given task. Each algorithm has credits and demerits in different situations. If you have a million integer values between **-2147483648** and **+2147483647** and you need to sort them, the bin sort is the accurate algorithm to use. If you have a million book titles, the quick sort algorithm might be the best choice. By knowing the toughness and weaknesses of the different algorithms, you pick the best one to accomplish a specific task or to solve a specific problem.

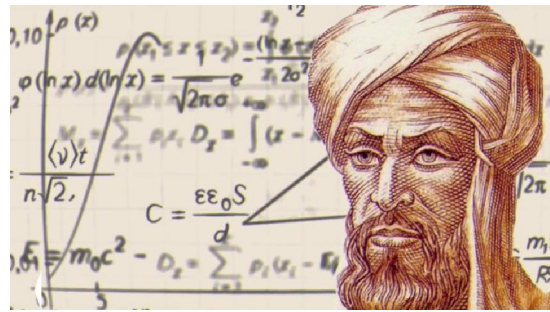
One of the most important aspects of an algorithm is how fast it can manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item. It is often easy to come up with a list of rules to follow in order to solve a problem, but if the algorithm is too slow, it's back to the drawing board. Efficiency of an algorithm depends on its design and implementation. Since every procedure or formula for solving a problem based on conducting a sequence of specified actions -- uses computer resources to run -- execution time and internal memory usage are important considerations to analyze an algorithm.



Why Study Algorithms?

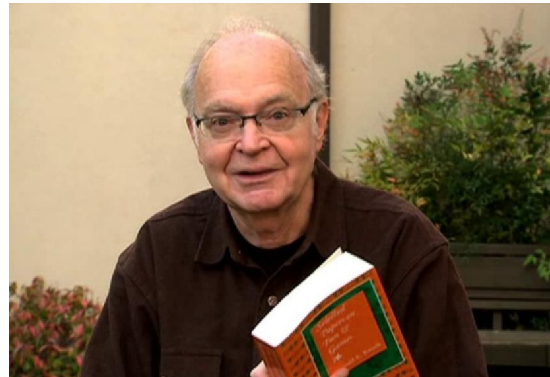
Algorithms are the heart of computer science (usually means a procedure or basically instance of logic written in software that solves a recurrent problem of finding an item with specific properties among collection of items or transforming data according to specified actions to protect it), and the subject has countless practical applications as well as intellectual depth that is widely used throughout all areas of information technology including solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that often involves repetition of an operation. The word algorithm -- a mathematical concept whose roots date

back to 600 AD with invention of the decimal system -- derives from the name of the ninth century Persian mathematician and geographer.



Mohammed ibn-Musa al-Khwarizmi, who was part of the royal court in Baghdad and who lived from about 780 to 850. On the other hand, it turns out algorithms (widely recognized as the foundation of modern computer coding) have a long and distinguished history stretching back as far as the Babylonians.

Although there is some available body of facts or information about early multiplication algorithms in Egypt (around **1700-2000 BC**) the oldest algorithm is widely recognized to be valid or correct to have been found on a set of **Babylonian clay tablets** that date to around **1600 - 1800 BC**. Their exact significance only came to be revealed or exposed around 1972 when an American computer scientist, mathematician, and professor emeritus at Stanford University .



Donald E. Knuth published the first English translations of various Babylonian **cuneiform** mathematical tablets.

Here are some short extracts from his 1972 manuscript that explain these early algorithms:-

"The calculations described in Babylonian tablets are not merely the solutions to specific individual problems; they are actually general procedures for solving a whole class of problems." - Pages 672 to 673 of "Ancient Babylonian Algorithms".

The wedge-shaped marks on clay tablets also seem to have been an early form of instruction manual:-

"Note also the stereotyped ending, 'This is the procedure,' which is commonly found at the end of each section on a table. Thus the Babylonian procedures are genuine algorithms, and we can commend the Babylonians for developing a nice way to explain an algorithm by example as the algorithm itself was being defined...." - Pages 672 to 673 of "Ancient Babylonian Algorithms".

The use of computers, however, has raised the use of algorithms in daily transactions (like accessing an automated teller machine (ATM), booking an air or train or buying something online) to unprecedented levels of real-world problems with solutions requiring advanced algorithms abounds. From **Google search** to morning routines, algorithms are ubiquitous in our everyday life -- and their use is only likely to grow to break down tasks into chunks that can be solved through specific implementations. Many of the problems, though they may not seem realistic, need the set of well-defined algorithmic knowledge that comes up every day in the real world. By developing a good understanding of a series of logical steps in an algorithmic language, you will be able to choose the right one for a problem and apply it properly. Different algorithms play different roles in programming – and algorithms are used by computer programs **where a program –**

- Get input data.
- Process it using the complex logics.
- Stop when it finds an answer or some conditions are met.
- Produce the desired output.

To give you a better picture, here is the most common type of algorithms:

- Searching Algorithms
- Sorting Algorithms
- Path finding Algorithms
- Tree and graph based algorithms
- Approximate Algorithms
- Compression Algorithms
- Random Algorithms
- Pattern Matching

- Sequence Finding and a lot more

You only need to define your problem then select the right algorithm to use. The word **algorithm** may not appear closely connected to kids, but the truth is that -- for kids -- understanding the process of building a step by step method of solving a problem helps them build a strong foundation in logical thinking and problem solving. Here are some problems you can ask your kid to discuss algorithmic solutions with you:

- How do we know if a number is odd or even?
- How do we calculate all of the factors of a number?
- How can we tell if a number is prime?
- Given a list of ten numbers in random order, how can we put them order?

Algorithms has shown it can yield results in all industries — from predicting insurance sales opportunities and generating the millions of search inquiries every day to automating medicine research, optimizing transportation routes, and much more. While algorithms help companies like **Master Card** and **Visa** to keep their users' information, such as card number, password, and bank statement safely -- **algorithms aren't perfect**. They fail and some fail spectacularly. Over the past few years, there have been some serious fails with algorithms, which are the formulas or sets of rules used in digital decision-making processes. Now people are questioning whether we're putting too much trust in the algorithms. When algorithms go bad: Online failures show humans are still needed. Disturbing events at **Facebook**, **Instagram** and **Amazon** reveal the importance of context.

References

1. What's the Importance of Algorithms in Computer Programming? By Cleophas Mulongo.
2. 15 of the Most Important Algorithms That Helped Define Mathematics, Computing, and Physics By Christopher McFadden.
3. The Importance of Algorithms for Kids | Juni Learning.
4. Algorithms By Jeff Erickson.