# What is an Algorithm?

## History

The word algorithm comes from the name of a persian author, **Abu Ja'far Mohammed ibn Musa al Khowarizmi** (c. 825 A.D), who wrote a book on mathematics. This word has taken on a special significance in computer science, where algorithm has come to refer as a method that can be used by a computer for the solution of a problem. This what makes algorithm different from words such as process, technique or method.

## Definition

An algorithm is a finite set of instructions that are designed to accomplishes a paticular task.

## What algorithm means in programming?

So, what is a programming algorithm? you can think of a programming algorithm as a recipe that describes the exact steps needed for the computer to solve problem. We've all seen a food recipes- they list the ingredients needed and a set of steps for how to make the described meal. Well, an algorithm is just like that. In computing world, the word for a recipe is a **procedure** and the ingredients are called **Inputs** and final meal is the **output**. A programming algorithm describes how to do something, and your computer will do it exactly that way every time. However, it's important to note that a algorithm is not a computer code. It's written in simple english (or whatever the programmers speaks).

## How to write a good algorithm?

To, write a good algorithm you must have to follow some criteria before writting:

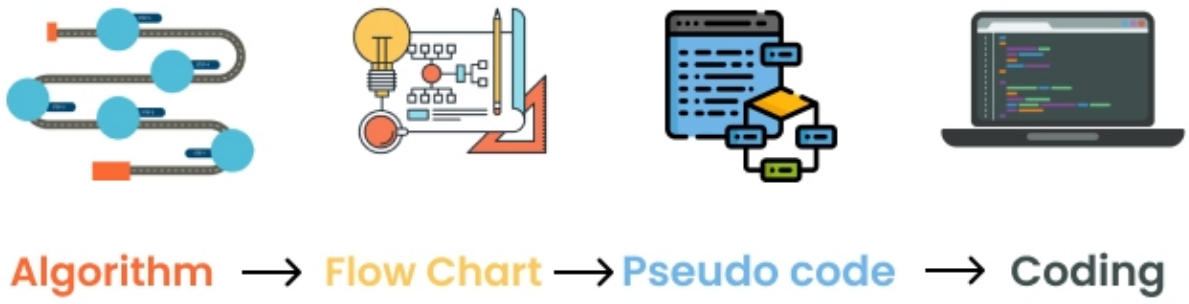
- **Input and output must be specified-** An input is the data transferred by the user to the program to produce certain output. An algorithm can have 0 or more inputs from the user, If the inputs must be taken from the user the details must be specified in the algorithm. An output is the result that is generated by the program according to the given instructions for the given input. Note that the algorithm must produce at least one output.

- **All important steps must be mentioned-** An algorithm is a step-by-step process that is performed in a program, thus every important step must be present in the algorithm in considerate details.
- **Instructions must be cleared and perfectly ordered-** Each instruction should be clear and unambiguous. ordering the instructions in perfect order will help the program to accomplish a paticular task.
- **Instructions should have to be effective-** Every instruction must be very basic so that it can be carried out, in principle, by a person using only pencil and paper in a finite amount of time. Performing arithmetic on integers is an example of an effective operation, but arithmetic with real is not, since some value must be expressible only bu infinitely long decimal expansion. Adding two such numbers would violate the effectiveness property.
- **The Algorithm must contain finite number of steps-** If we trace out the instructions of an algorithm, then for all cases, the algorithm terminates after a finite number of steps.

## Purpose of algorithm

Writing Algorithms of the program before start coding is a very beneficial habit in computer programming. It resolves various errors, confusion while programming for big projects.

Algorithms provide a better understanding towards programming and helps in building better problem solving logic. Algorithms help the programmer to write code faster by following steps in Algorithm.




### How do we actually write a algorithm?

1. **Addition of two numbers (numbers are entered by user)**

```
Step 1: Start.
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values of num1 and num2.
Step 4: Add the values of num1 and num2 and assign the result to variable sum.
Step 5: Display sum
Step 6: End
```

2. **Comparison of 3 numbers to find the largest number**

```
Step 1: Start.
Step 2: Declare variables num1, num2 and num3.
Step 3: Read values of num1, num2 and num3.
Step 4: Compare num1, num2 and num3
If num1 > num2
            If num1 > num3
                        Display num1 is the largest number
            Else
                        Display num2 is the largest number
Else
            If num2 > num3
                        Display num2 is the largest number
            Else
                        Display num3 is the largest number
Step 5: Display sum
Step 6: End
```

## What kinds of problems are solved by algorithms?

Practical applications of algorithms are everywhere and the following are some examples that will show you the daily life problems that solve by an algorithm.

- The Internet enables people all around the world to quickly acess and retrieve large amount of information. With help of efficient algorithm technique like- finding good routes on which data will travel, and using a search engine to quickly find pages on which paticular information resides.
- Electronic commerce enables goods and services to be negotiated and exchanged electronically, and it depends on the privacy of personal information such as credit card numbers, passwords and bank statements. The core technologies used in electronic commerce include public-key cryptography and digital signatures, which are based on numerical algorithms and number theory.
- We are given a road map on which the distance between each pair of adjacent intersections is marked, and we wish to determine the shortest route from one intersection to another. The number of possible routes can be huge, even if we

disallow routes that cross over themselves. How do we choose which of all possible routes is the shortest? Here, we model the road map as a graph and we wish to find the shortest path from one vertex to another in the graph.

# Types of algorithm (Algorithm design techniques)

Algorithms are classified into many types according to it’s functionality and use. Here are some of the most important algorithms.

## Divide and Conquer Algorithm

Divide and Conquer is an algorithm paradigm. A Divide and Conquer algorithm repeatedly breaks down a problem into sub-problems and solves the sub-problem.

The sub-problems are the related problems of the original problem. The solution to the sub-problems is merged to provide solution to the original problem.

## Dynamic Programming Algorithm

Dynamic Programming is an algorithm design paradigm. It is mainly used in problems involving optimization and overlapping.

## Greedy Algorithm

Greedy Algorithm is a simple and effective algorithm. It is mainly used in problems involving optimization and it uses the strategy to solve the problem that seems best.

## Backtracking Algorithm

Backtracking Algorithm is a general algorithm used for solving computational problems. It uses recursion to solve the problem

# Role of algorithm in programming

## Factor that determine the program performance

### Primary factor

- Algorithms
- Data Structures

### Secondary factor

- Hardware

- Compiler
- Programming language
- Operating System
- Programmer Effort

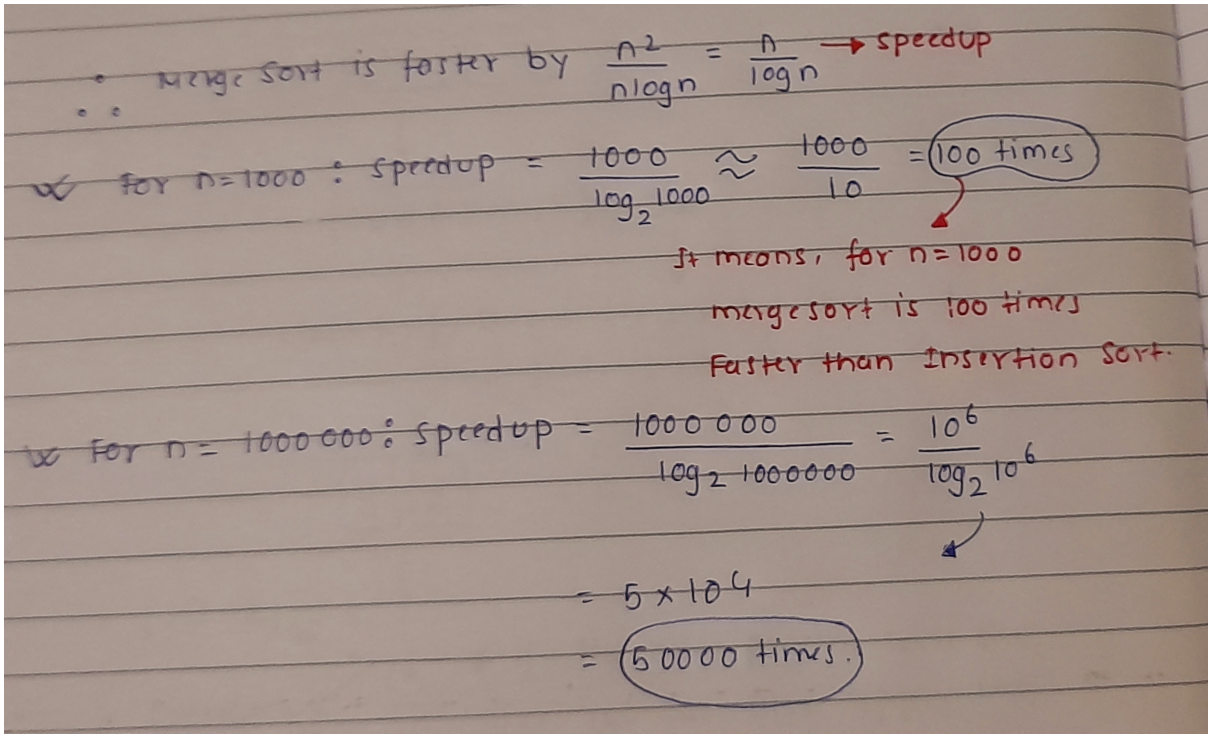
**Note-** There could be many factors that applies to an algorithm but here I mentioned some common factors only.

## Why algorithms are major factor in programming?

Consider the below example to understand better why algorithms are major factor in programming.

**Example-**

Consider Merge Sort with running time O(nlogn) compared to Insertion Sort with running time O(n^2).




The example shows why primary factors are more important than secondary factors. Here, we can see that two famous sorting algorithms Merge Sort and Insertion Sort are getting compared with their running time and we can see that Merge Sort is **(n/logn) times** faster than Insertion Sort.

Now, when you start taking Inputs(n) you can see the vast running time difference between both the algorithms.

**Consider-**

- for n = 1000 (Merge Sort is 100 times faster then Insertion sort)
- for n = 1000000 (Merge Sort is 5000 times faster then Insertion sort)

So, when you look up to the factors (Secondary factors), you will see that none of the factors is going to give a speedup of 100 times, 50000 times etc.

**Example-**

Suppose, you have created a program that runs on both Windows 7 and Windows 10. Now, If I ask you which program will run faster? offcourse your answer will be, program that created on Windows 10 operating system will run faster than the Windows 7 one.

Now, the point is that, the program in Windows 10 will be 2-4 times faster than the program on Windows 7, It will not going to be thousand or millions times faster. The running time difference between both the operating system should be in order like (2 or 3 or 4) times.

**Why algorithms matter to our program more than other factors?**

You can see that running a program on good devices will not going to achieve that much of speed. To, achieve enormous speed for your program, then your have to Implement some efficient algorithms and data structures to your program that is capable of running to any computer whether it’s good or bad.