



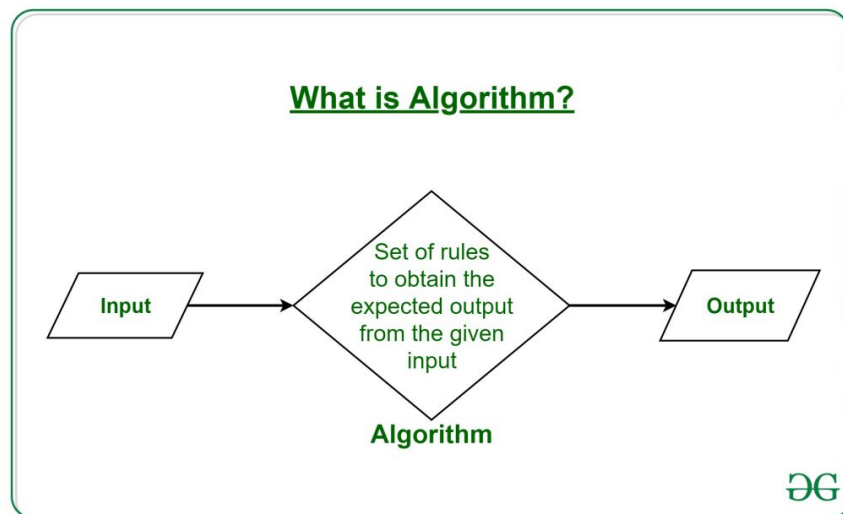
Algorithm



The word Algorithm means "A set of finite rules or instructions to be followed in calculations or other problem-solving operations"

Or

"A procedure for solving a mathematical problem in a finite number of steps that frequently involves recursive operations".



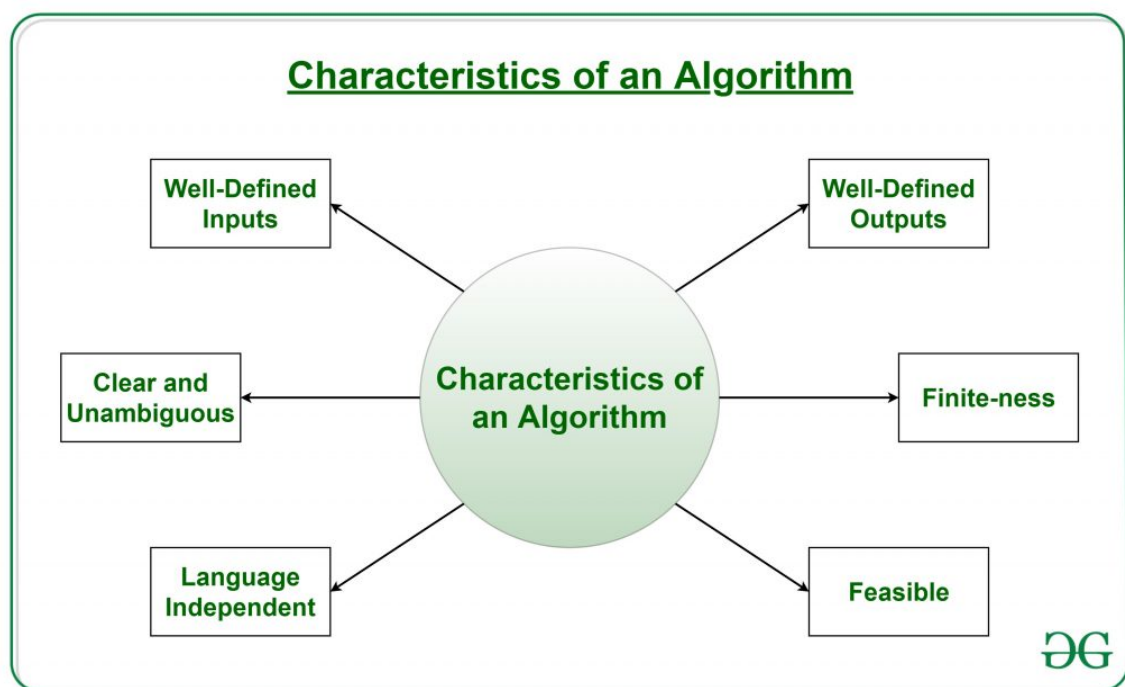
Algorithms can be simple and complex depending on what you want to achieve.

What is the need for algorithms?

1. Algorithms are necessary for solving complex problems efficiently and effectively.
2. They help to automate processes and make them more reliable, faster, and easier to perform.
3. Algorithms also enable computers to perform tasks that would be difficult or impossible for humans to do manually.
4. They are used in various fields such as mathematics, computer science, engineering, finance, and many others to optimize processes, analyze data, make predictions, and provide solutions to problems.

Properties of Algorithm:

1. It should terminate after a finite time.
2. It should produce at least one output.
3. It should take zero or more input.
4. It should be deterministic means giving the same output for the same input case.
5. Every step in the algorithm must be effective i.e. every step should do some work.



Types of Algorithms:

1. Brute Force Algorithm:

It is the simplest approach to a problem. A brute force algorithm is the first approach that comes to finding when we see a problem.

2. Recursive Algorithm:

A recursive algorithm is based on recursion. In this case, a problem is broken into several sub-parts and called the same function again and again.

3. Backtracking Algorithm:

The backtracking algorithm builds the solution by searching among all possible solutions. Using this algorithm, we keep on building the solution following criteria. Whenever a solution fails we trace back to the failure point build on the next solution and continue this process till we find the solution or all possible solutions are looked after.

4. Search Algorithm:

Searching algorithms are the ones that are used for searching elements or groups of elements from a particular data structure. They can be of different types based on their approach or the data structure in which the element should be found.

5. Sorting Algorithm:

Sorting is arranging a group of data in a particular manner according to the requirement. The algorithms which help in performing this function are called sorting algorithms. Generally sorting algorithms are used to sort groups of data in an increasing or decreasing manner.

6. Hashing Algorithm:

Hashing algorithms work similarly to the searching algorithm. But they contain an index with a key ID. In hashing, a key is assigned to specific data.

7. Divide and Conquer Algorithm:

This algorithm breaks a problem into sub-problems, solves a single sub-problem, and merges the solutions to get the final solution. It consists of the following three steps:

- Divide

- Solve
- Combine

8. Greedy Algorithm:

In this type of algorithm, the solution is built part by part. The solution for the next part is built based on the immediate benefit of the next part. The one solution that gives the most benefit will be chosen as the solution for the next part.

9. Dynamic Programming Algorithm:

This algorithm uses the concept of using the already found solution to avoid repetitive calculation of the same part of the problem. It divides the problem into smaller overlapping subproblems and solves them.

10. Randomized Algorithm:

In the randomized algorithm, we use a random number so it gives immediate benefit. The random number helps in deciding the expected outcome.

How to Design an Algorithm?

To write an algorithm, the following things are needed as a pre-requisite:

1. The **problem** that is to be solved by this algorithm i.e. clear problem definition.
2. The **constraints** of the problem must be considered while solving the problem.
3. The **input** to be taken to solve the problem.
4. The **output** is to be expected when the problem is solved.
5. The **solution** to this problem is within the given constraints.

References

GeeksforGeeks | A computer science portal for geeks

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles, quizzes and practice/competitive

 <https://www.geeksforgeeks.org/>



Author → Serhat Kumas

<https://www.linkedin.com/in/serhatkumas/>

SerhatKumas - Overview

Computer engineering student who loves coding in different fields instead of focusing on a one spesific area. - SerhatKumas

 <https://github.com/SerhatKumas>

