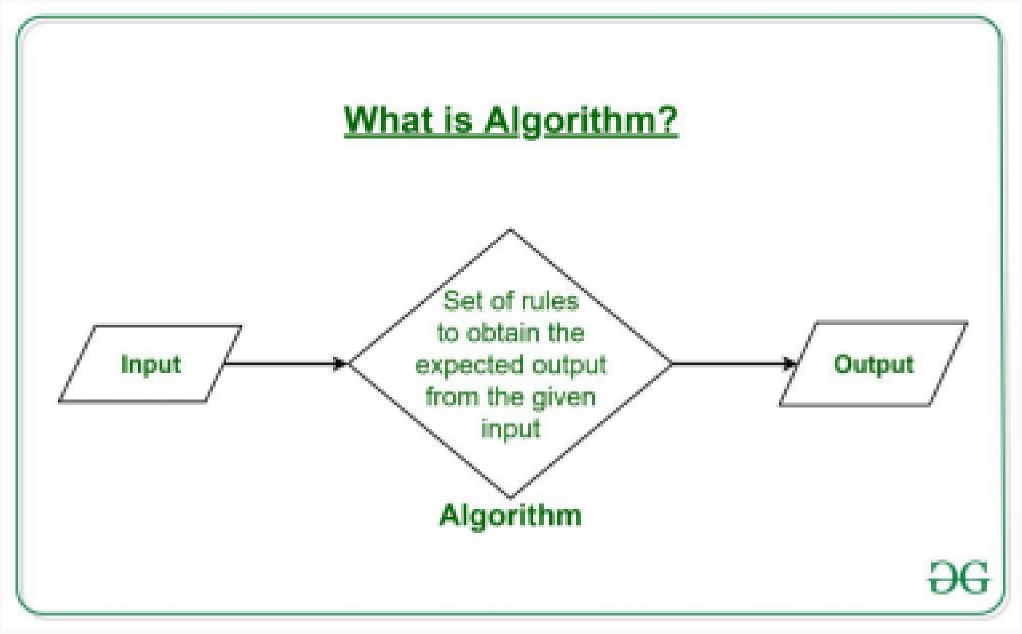
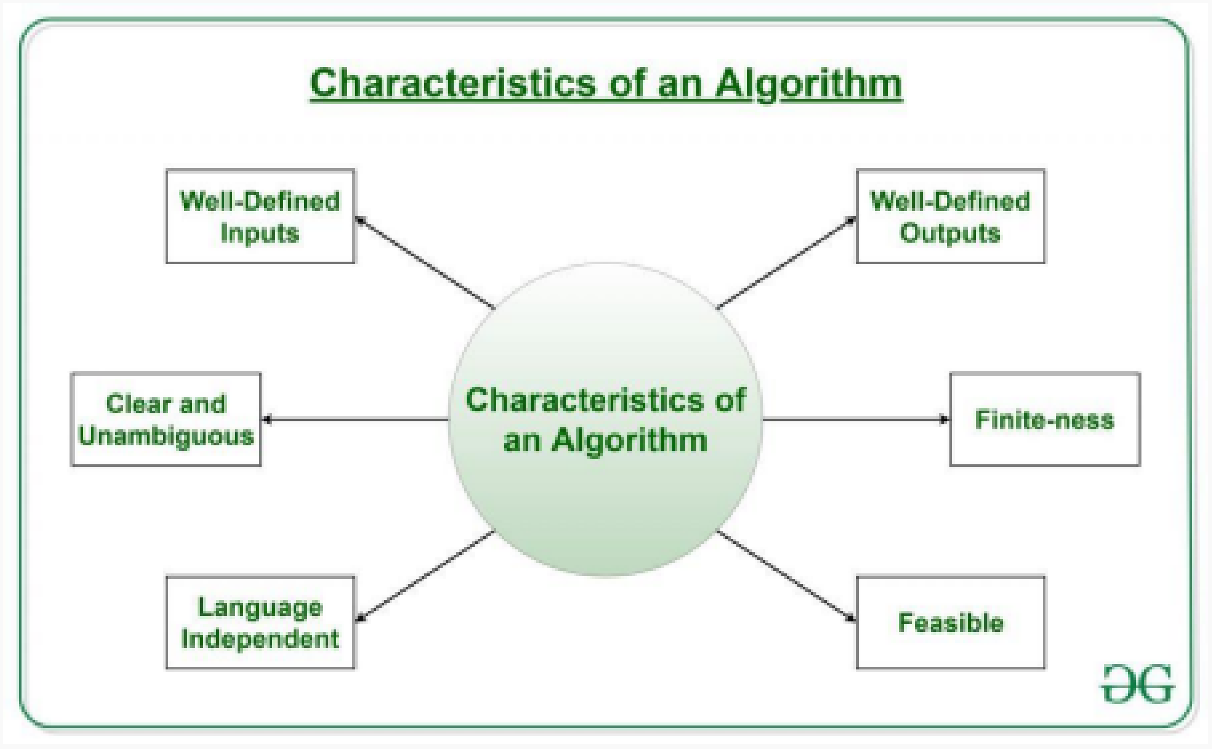
Introduction to Algorithms

Algorithm refers to a sequence of finite steps to solve a particular problem

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”.


What is Algorithm?
Input
Set of rules to obtain the expected output from the given input
Algorithm
Output



Use of the Algorithms:

Algorithms play a crucial role in various fields and have many applications. Some of the key areas where algorithms are used include:

Computer Science: Algorithms form the basis of computer programming and are used to solve problems ranging from simple sorting and searching to complex tasks such as artificial intelligence and machine learning.

Mathematics: Algorithms are used to solve mathematical problems, such as finding the optimal solution to a system of linear equations or finding the shortest path in a graph.

Operations Research: Algorithms are used to optimize and make decisions in fields such as transportation, logistics, and resource allocation.

Artificial Intelligence: Algorithms are the foundation of artificial intelligence and machine learning, and are used to develop intelligent systems that can perform tasks such as image recognition, natural language processing, and decision-making.

Data Science: Algorithms are used to analyze, process, and extract insights from large amounts of data in fields such as marketing, finance, and healthcare.

What is the need for algorithms?

Algorithms are necessary for solving complex problems efficiently and effectively.

Properties of Algorithm:

- It should terminate after a finite time.
- It should produce at least one output.
- It should take zero or more input.
- It should be deterministic means giving the same output for the same input case.

Every step in the algorithm must be effective i.e. every step should do some work.

Types of Algorithms:

There are several types of algorithms available. Some important algorithms are:

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. Searching 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.
To learn more about the types of algorithms refer to the article about “Types of Algorithms“.

Advantages of Algorithms:

• It is easy to understand.
• An algorithm is a step-wise representation of a solution to a given problem. • In an Algorithm the problem is broken down into smaller pieces or steps hence, it is easier for the programmer to convert it into an actual program.

Disadvantages of Algorithms:

• Writing an algorithm takes a long time so it is time-consuming. • Understanding complex logic through algorithms can be very difficult. • Branching and Looping statements are difficult to show in Algorithms(imp). • How to Design an Algorithm?
• To write an algorithm, the following things are needed as a pre-requisite: • The problem that is to be solved by this algorithm i.e. clear problem definition. • The constraints of the problem must be considered while solving the problem. • The input to be taken to solve the problem.
• The output is to be expected when the problem is solved.
• The solution to this problem is within the given constraints.

- Then the algorithm is written with the help of the above parameters such that it solves the problem.

Example: Consider the example to add three numbers and print the sum.

Step 1: Fulfilling the pre-requisites

As discussed above, to write an algorithm, its prerequisites must be fulfilled.

The problem that is to be solved by this algorithm: Add 3 numbers and print their sum.

The constraints of the problem that must be considered while solving the problem: The numbers must contain only digits and no other characters.

The input to be taken to solve the problem: The three numbers to be added.

The output to be expected when the problem is solved: The sum of the three numbers taken as the input i.e. a single integer value.

The solution to this problem, in the given constraints: The solution consists of adding the 3 numbers. It can be done with the help of the '+' operator, or bit-wise, or any other method.

Step 2: Designing the algorithm

Now let's design the algorithm with the help of the above pre-requisites:

Algorithm to add 3 numbers and print their sum:

```
START
Declare 3 integer variables num1, num2, and num3.
Take the three numbers, to be added, as inputs in variables num1, num2, and num3 respectively.
Declare an integer variable sum to store the resultant sum of the 3 numbers.
Add the 3 numbers and store the result in the variable sum.
Print the value of the variable sum
END
```