**Definition:**

An **algorithm** is a **finite, well-defined sequence of instructions** or steps designed to solve a specific problem or perform a computation.

More formally, an algorithm has the following key properties:

1. **Input**: It accepts zero or more inputs.
2. **Output**: It produces at least one output.
3. **Definiteness**: Each step is precisely and unambiguously defined.
4. **Finiteness**: The algorithm must terminate after a finite number of steps.
5. **Effectiveness**: All operations must be basic enough to be carried out exactly and in a finite amount of time.

| **Role** | **Description** |
| --- | --- |
| **1. Automation** | Algorithms provide a clear set of steps that can be executed by a computer without human intervention. |
| **2. Efficiency** | Choosing or designing the right algorithm allows for solving problems faster and using fewer resources. |
| **3. Scalability** | Algorithms help solve problems that are too large or complex for manual solving, such as sorting millions of records or processing real-time data. |
| **4. Reusability** | Once an algorithm is designed, it can often be reused for similar types of problems (e.g., searching or sorting). |
| **5. Foundation for Programming** | Algorithms are the blueprint for writing code—programs are implementations of algorithms in specific languages. |

#### ****Examples of Algorithms:****

* **Searching Algorithm** – e.g., Binary Search
* **Sorting Algorithm** – e.g., Quick Sort, Merge Sort
* **Graph Algorithms** – e.g., Dijkstra’s shortest path
* **Machine Learning Algorithms** – e.g., Decision Trees, K-Means Clustering