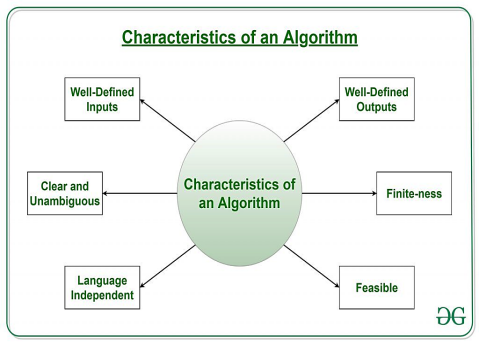
# - What is an algorithm? Provide a formal definition.

* An algorithm is a step-by-step procedure or set of rules to solve a problem or perform a task in a finite number of steps.

1. - List and explain the key characteristics of an algorithm.



* **Finite:** Must terminate after a finite number of steps.
* **Definite:** Each step must be clearly defined.
* **Input:** Can have zero or more inputs.
* **Output:** Must produce at least one output.
* **Deterministic:** Same input always gives the same output.
* **Effective:** Every step must perform a meaningful task.

1. - Why is it important for an algorithm to be deterministic?

* **Ensures consistency**: The same input always produces the same output.
* **Predictability:** Helps in debugging and verifying correctness.

1. - Discuss the applications of algorithms .

* **Computer Science:** Used in programming, AI, and machine learning.
* **Mathematics**: Solves linear equations, finds shortest paths in graphs.
* **Operations Research:** transportation, logistics, and resource allocation.
* **Data Science:** Analyzes, processes, marketing, finance, and healthcare.

1. - How are algorithms used in Mathematics/computer science? Provide examples.

**(Applications in computer science**)

* **Programming:** Writing efficient code.
* **AI & Machine Learning:** Training models, decision-making.
* **Data Structures:** Sorting, searching, and organizing data.

**(** Applications in **Mathematics)**

* **Linear Algebra:** Solving systems of equations.
* **Graph Theory:** Finding the shortest path (e.g., Dijkstra's algorithm).
* **Optimization:**Maximizing or minimizing functions.

1. How to Design an Algorithm?

**Design:** Focuses on creating algorithms to solve a specific problem.

* **Understanding the Problem**: Define the problem clearly.
* **Choosing a Strategy:** Select an approach (e.g., divide and conquer).
* **Developing the Algorithm:** Write step-by-step logic.
* **Verification:** Ensure correctness with test cases.

1. How to Analyze an Algorithm?

**Analysis:** Focuses on evaluating the performance and efficiency of an algorithm.

* **Correctness**: Ensures the algorithm produces the right output.
* **Complexity Analysis:** Evaluates time and space efficiency.
* **Best/Worst/Average Case Analysis:** Measures performance in different scenarios.
* **Comparative Analysis:** Compares efficiency with other algorithms.

1. - Explain the concept of time complexity and space complexity.

* **Time Complexity:** Measures how long an algorithm takes to run based on input size.
* **Space Complexity:** Measures how much memory an algorithm uses based on input size.

1. - What is the difference between best case, worst case, and average case analysis?

* **Best Case:** Minimum time/space required (ideal scenario, e.g., input is already sorted).
* **Worst Case:** Maximum time/space required (most challenging scenario, e.g., input is in reverse order).
* **Average Case:** Expected time/space required over all possible inputs (typical scenario).

1. - Define time complexity and provide an example of how it is calculated.

* **Time Complexity:** Measures the time taken by an algorithm as a function of input size.
* **Example:**A loop running **n**times has a time complexity of**O(n)**.

1. - Define space complexity .

* **Space Complexity:**Measures the amount of memory an algorithm uses as a function of input size.
* Includes fixed space and variable space (e.g., dynamic memory).

1. - What are the advantages/dis of using algorithms in problem-solving?

* Easy to understand
* Serve as a blueprint
* Simple to debug
* Not dependent on programming language

(DisAdvantage)

**Time & Space Complexity:** Some algorithms require too much time or memory.

**Difficult Representation:** Branching and looping logic can be hard to express.