

Introduction to Algorithms

An algorithm is a step-by-step procedure or formula for solving a problem. It is a finite sequence of instructions or logic, written to accomplish a certain predefined task. Algorithms are fundamental to computer science and programming, forming the basis for software development.

Characteristics of an Algorithm:

1. Finiteness: An algorithm must always terminate after a finite number of steps.
2. Definiteness: Each step of the algorithm must be precisely defined.
3. Input: An algorithm should have zero or more inputs.
4. Output: It must produce at least one output.
5. Effectiveness: All operations should be basic and directly executable.

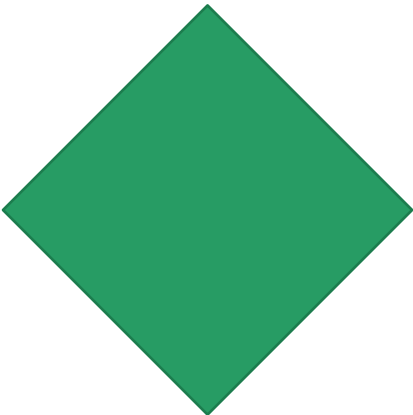
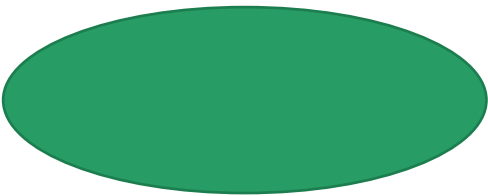
Applications:

- Sorting (e.g., Bubble Sort, Merge Sort)
- Searching (e.g., Binary Search)
- Data structure manipulation (e.g., Tree traversal)
- Optimization problems

Introduction to Flowcharts

A flowchart is a graphical representation of an algorithm or process. It uses various symbols to represent different types of actions or steps in a process, connected by arrows to indicate the flow of control.

Common
Flowchart Symbols:



1. Oval: Represents the start or end of a process.
2. Rectangle: Denotes a process step or instruction.
3. Diamond: Represents a decision point in the process.
4. Arrow: Shows the flow of control.

Advantages of Flowcharts:

- Helps visualize the process clearly.
- Simplifies complex processes for better understanding.
- Facilitates debugging and maintenance.

Limitations:

- Can become cumbersome for large and complex processes.
- Modifications may require redrawing.

Comparison and Importance

Comparison between Algorithm and Flowchart:

1. Representation

- Algorithm: Represented in textual form using pseudo-code or structured English.
- Flowchart: Represented graphically using symbols and arrows.

2. Simplicity

- Algorithms are easier for a programmer to understand and modify.
- Flowcharts are more intuitive for non-programmers.

Importance in Problem-Solving:

Both algorithms and flowcharts are crucial in planning, debugging, and maintaining code. They help break down problems into manageable steps, enabling clear communication and systematic solutions.

In conclusion, while algorithms focus on the logic and sequence of steps, flowcharts emphasize the visual understanding of a process