Algorithms

- A finite set of instructions which, if followed, accomplish a particular task.
- An algorithm is the step-by-step unambiguous instructions to solve a given problem.
- Typically, an algorithm takes an input data and produces an output based upon it.

Let us consider the problem of preparing an omelette. To prepare an omelette, we follow the steps given below:

1) Get the frying pan.
2) Get the oil.
a. Do we have oil?
i. If yes, put it in the pan.
ii. If no, do we want to buy oil?
1. If yes, then go out and buy.
2. If no, we can terminate.
3) Turn on the stove, etc...

Expressing an Algorithm

- **Natural Language-** verbose and ambiguous and are rarely used for complex or technical algorithm (e.g., English or French)
- **Pseudocode** – English-like syntax that resembles a programming language
- **Flowchart** – graphical representation of an algorithm
- **Programming language** – expressing algorithm in a form that can be executed by a computer (e.g., FORTRAN, C, Pascal, Java, PHP, etc.)

Algorithm Criteria

- **Input**: there are zero or more quantities which are externally supplied;
- **Output**: at least one quantity is produced;
- **Definiteness**: each instruction must be clear and unambiguous;
- **Finiteness**: if we trace out the instructions of an algorithm, then for all cases the algorithm will terminate after a finite number of steps;
- **Effectiveness**: every instruction must be sufficiently basic that it can in principle be carried out by a person using only pencil and paper. It is not enough that each operation be definite, but it must also be feasible.

Fundamental questions about algorithms

Given an algorithm to solve a particular problem, we are naturally led to ask:

1. What is it supposed to do? (Specification)
2. Does it really do what it is supposed to do? (Verification)
3. How efficiently does it do it? (Performance Analysis)

Why the Analysis of Algorithms?

In computer science, multiple algorithms are available for solving the same problem (for example, a sorting problem has many algorithms, like insertion sort, selection sort, quick sort and many more). Algorithm analysis helps us to determine which algorithm is most efficient in terms of time and space consumed.

Goal of the Analysis of Algorithms

The goal of the analysis of algorithms is to compare algorithms (or solutions) mainly in terms of running time but also in terms of other factors (e.g., memory, developer effort, etc.)