Introduction to data structure and algorithms (Under Course CS1302)

Data: Facts and figures collected through observations, counting, measurements, etc.

Data Type: Refers to the domain (set) from which values of data may be taken and set of operations that can be performed over those values.

For example: Integer refers to set of mathematical integer values. Allowed operations on integers *x* and *y* are addition, subtraction, multiplication, division, modulus (5 mod 3 or x % y), successor(x), predecessor(y), etc.

Float refers to set of mathematical integer as well as fractional values. Allowed operations on float *x* and *y* are addition, subtraction, multiplication, division. Note that unlike integers, float does not support the operations modulus (x mod y), successor, and predecessor.

Boolean refers to a data that have a domain with only two values {True, False} which may sometimes be represented as {1, 0} too. Allowed operations over boolean data are logical x AND y, x OR y, x

XOR y, NOT x, etc. It does not support algebraic addition, multiplication, etc.

Data Structure: It refers to techniques of organizing a set of data elements in the computer's memory for efficient storage, processing and retrieval.

- Term efficient refers to using less time and memory. Sometimes high data security may also be considered.

- Some examples of Data Structures are Array, Linked List, Stack, Queue, Tree, Graph, etc.

- Data Structures are key elements for writing any computer program and building software.

Algorithm:

A computer algorithm is a finite sequence of instructions (known basic steps) to solve a particular type of problem.

In real world, examples of algorithm are

- A cooking recipe to cook a particular food.
- A user manual for washing cloths using a washing machine
- Sequence of steps followed to wear a tie
- Sequence of steps for assembling and installing a ceiling fan
- Sequence of steps for assembling a new PC using its discrete components
- Steps for hot pressing a washed and dried T-shirt using an electric iron
- Steps for changing the punctured tire of a car/scooter
- Sequence of steps to install Linux and Windows both in a new hard disk in order to

prepare a PC or laptop for dual booting system

Properties of Algorithms:

Input: It may require zero or more input.

Output: It must produce one or more output.

Definiteness: Algorithm steps must be clear and unambiguous.

If there are multiple knobs in a washing machine, then an instruction asking to turn the knob is unambiguous.

The instruction must clearly specify which specific knob should be turned, in which direction clockwise or counter clockwise and by how much.

Finiteness: For all possible input data, sequence of steps of an algorithm must terminate after a finite number of steps.

Any algorithm cannot have endless number of steps to before it outputs the intended result.

Effectiveness: Steps of algorithms should be sufficiently basic so that target person or computer may comfortably carry out those steps.

In a cooking recipe, there cannot be a step asking to heat the oil or water up to 70^0 C, because there is no precise temperature measuring system available in any normal kitchen.

In a normal travel direction, it is not expected to have a step to swim across a river or jump over a wall or fence.

What to study about algorithms?

1. Devising an algorithm to solve the problem at hand.

2. Expressing the devised algorithm using
    a. Sequence of English language steps
    b. Flowcharts
    c. Pseudo codes

3. Checking for correctness: Give logical or mathematical proof of correctness to ensure that the algorithm correctly performs the intended computation and gives correct output for all possible input.

4. Analyzing for efficiency: Computing time and space (memory) complexities with respect to input data size. Complexity refers to how there is increase in time/temporary memory requirements with respect to increase in input size.

   Complexity analysis is to investigate whether, the time/memory requirements are fixed (independent of input size) or a linear or quadratic or logarithmic or exponential or any other function of the input data size.

   For example, In sorting of *n* data items takes time T(n) = cn time, or cn^2 time or $cn\log n$ time or cn^k time or ce^n time, where *c* and *k* are constant values.

   Temporary memory requirement (space complexity) is analyzed in similar ways as the time requirement (time complexity).

We use O, Ω and θ notations to express the time/space complexities. Details of these notations would be explained in the upcoming classes.

5. Implement the Algorithm using a programming language like C++, Java, Python, MATLAB, etc and execute the program with various test data as input and verify the output for correctness.

6. Statistical Analysis:

   a. Execute the implemented algorithm for various data sets of different kinds and volumes and analyze the performance in for various types of inputs.

   b. Compare the performance of the algorithm with that of other algorithms devised to solve the same problem.