Introduction To Data Structures

ALGORITHMS & DATA STRUCTURES – I

COMP 221

Introduction To Data Structures

- Goal: to **organize data**
- Criteria: to facilitate **efficient**
  - **storage** of data
  - **retrieval** of data
  - **manipulation** of data
- Design Issue:
  - **select and design** appropriate data types. (This is the real essence of OOP.)

Data Structure

A ***data structure*** is a systematic way of organizing and accessing data.

In computer science, a data structure is a particular way of storing and organizing data in a computer memory so that it can be used efficiently.

Importance of Data Structures

1. **Solving problems with computers invariably requires working with**
   - **Data structures** to represent the data
   - **Algorithms** for processing the data
2. **For many applications, the choice of the proper data structure is the only major decision**
   - Once the choice has been made, the necessary algorithm is simple
   - For the same data, some data structures require more or less space than others **(save space)**
   - For the same operations on the data, some data structures lead to more or less efficient algorithm than others **(save time)**

Classification of Data Structures

Linear

- **Pointer**
- **Array**
- **Stack**
- **Queue** (Linear Queue, Circular Queue, Priority Queue)
- **Linked Lists** (Singly Linked List, Doubly Linked List)

Non Linear

- **Trees**
- **Graphs**

Data Types

- **Simple**
- **Structured — data structures** **— containers**

Algorithm

- May be written in pseudocode
- Characteristics of steps (instructions), see pg 9:
  - Definite and unambiguous
  - Simple
  - Finite
- Difference between correctness and efficiency, see pp 7-8
  - **O(n)** — grows linearly with size (n) of the input
  - **O(1)** — is constant , i.e. independent of size of input
- Should be well-structured

Can't separate data structures and algorithms

Algorithms + Data Structures = Programs

- An ***algorithm*** is a step-by-step procedure for solving a problem in a finite amount of time.

Examples of Algorithms:

Travel instructions

Cooking recipes

a car manual page (how to remove the gearbox),

a medical procedure.

Algorithm Example

1. Accept an integer for *n*.
2. Initialize *sum to 0*.
3. For each integer i in the range 1 to *n:*

   Assign *sum + i to sum*.
4. Return the value of *sum*.

THANK YOU