

Introduction to Data Structure

Prepared by: Lec Tasmiah Tamzid Anannya, CS Dept, AIUB

Data & Structure

▶ What is *Data*?

- ▶ Data means raw facts or information that can be processed to get results.

▶ What is *Structure*?

- ▶ Some elementary items constitute a unit and that unit may be considered as a structure.
- ▶ A structure may be treated as a frame where we organize some elementary items in different ways.



Data Structure

- ▶ So, what is *Data Structure*?
 - ▶ Data structure is a structure where we organize elementary data items in different ways and there exists structural relationship among the items so that it can be used efficiently.
 - ▶ In other words, a data structure is means of structural relationships of elementary data items for storing and retrieving data in computer's memory.



Data Structure...

- ▶ Usually elementary data items are the **elements** of a data structure.
- ▶ However, a ***data structure may be an element of another data structure***. That means a data structure may contain another data structure.



Data Structure...

- ▶ We talk about/ study Data Structure in two ways:
 - ▶ Basic
 - ▶ Have a concrete implementation.
Example: Variable, Pointer, Array etc.
 - ▶ Abstract Data Types (ADTs):
 - ▶ ADTs are entities that are definition of data and operation but do not have any implementation. Example: List, Stack, Queue etc.



Data Structure...

- ▶ **Example of Data Structures:**

- ▶ Array, Linked List, Stack, Queue, Tree, Heap, Graph, Hash Table etc.

- ▶ **Types of elementary data item:**

- ▶ Character, Integer, Floating point numbers etc.

- ▶ **Elementary data item** - **Expression in C/C++**

- ▶ Character - **char**

- ▶ Integer - **int**

- ▶ Floating point number - **float**

- ▶ etc.



Operations on Data Structure

▶ Basic:

- ▶ *insertion* (addition of a new element in the data structure)
- ▶ *deletion* (removing the element from the data structure)
- ▶ *searching* (locating the element in the data structure)
- ▶ *Traversal* (accessing each data element in the data structure)

▶ Additional:

- ▶ *sorting* (Arranging elements in a data structure in a specified order)
- ▶ *merging* (combining elements of two similar data structures to create a new data structure of same type)
- ▶ etc.



Algorithm

- ▶ Set of **instructions** that can be followed to perform a **task**. In other words, **sequence of steps that can be followed to solve a problem**.
- ▶ To write an algorithm we do not strictly follow grammar of any particular programming language.
- ▶ However its language may be near to a programming language.



Program

- ▶ Sequence of **instructions of any programming language** that can be followed to perform a **particular task**.
- ▶ Like an algorithm generally a program has three sections such as **input, processing and output**.



Program...

- ▶ In a program usually we use a large amount of data. Most of the cases these data are not elementary items, where exists structural relationship between elementary data items.
 - ▶ *That means the programs uses data structure(s).*
- ▶ For a particular problem (usually for complex problem), at first we may write **an algorithm** then the algorithm may be converted into a **program**.



Complexity of algorithm

- ▶ **Two types of complexities:**

- ▶ Time Complexity

- ▶ Space Complexity



Time Complexity

- ▶ This complexity is related to *execution time* of the algorithm.
- ▶ Among many other factors, it mostly depends on the number of element *comparisons* and number of element *movement* (movement of data from one place to another).



Space Complexity

- ▶ This complexity is related to *space (memory) requirement* in the main memory for the data set used to implement the algorithm.
- ▶ That means if there are n data items used in an algorithm, the space complexity of the algorithm will be proportional to n .

