INTRODUCTION TO DATA STRUCTURES & ALGORITHMS

OMAR AL-MUKHTAR UNIVERSITY

FACULTY OF SCIENCE

COMPUTER SCIENCE DEPARTMENT

LECTURE 1

GRADING

- This course will include:
  - Three or Four programming written exercise's
  - Two or more Quizzes

- Grading:
  - Class participation: 10% → " homework's and quizzes"
  - Lab works: 20%
  - Midterm: 20%
  - Final: 50%

COURSE OUTLINE

- Course schedule:

Week	Topic
1	Introduction
2	STACK
3	Recursion
4 - 5	QUEUE
6 - 7	LINKED LISTS
8	Midterm
9 - 10	Trees
11	Graphs
12	Sorting Algorithms
13	Searching Algorithms
14	FINAL EXAM

WHAT IS DATA STRUCTURES

- Data structures and algorithms (DSA) are two essential concepts of any programming language.
- Every programming language has its own data structures and different types of algorithms to handle these data structures.

- Data Structure : a data structure is a way of storing and organizing data in computer memory so that it can be used (accessed & modified) efficiently.

- To design and implement Algorithms for developing a *Program* → *Organization* of DATA is IMPORTANT.

  - Logical or Mathematical Model of a Particular Organization of Data is Called

    >> DATA STRUCTURE <<

ALGORITHMS + DATA STRUCTURE = PROGRAMS

WHAT IS DATA STRUCTURES

- The *data structure* and the *operations* on organized data items can integrally solve the problem using a computer.

- DATA STRUCTURE = Organized data + Operations

WHY TO LEARN DATA STRUCTURES & ALGORITHMS ?

- As applications are getting complex and store a large amount of data , there are three common problems that applications face now-a-days.
  - 1. Data Search
  - 2. Speed
  - 3. Multiple requests

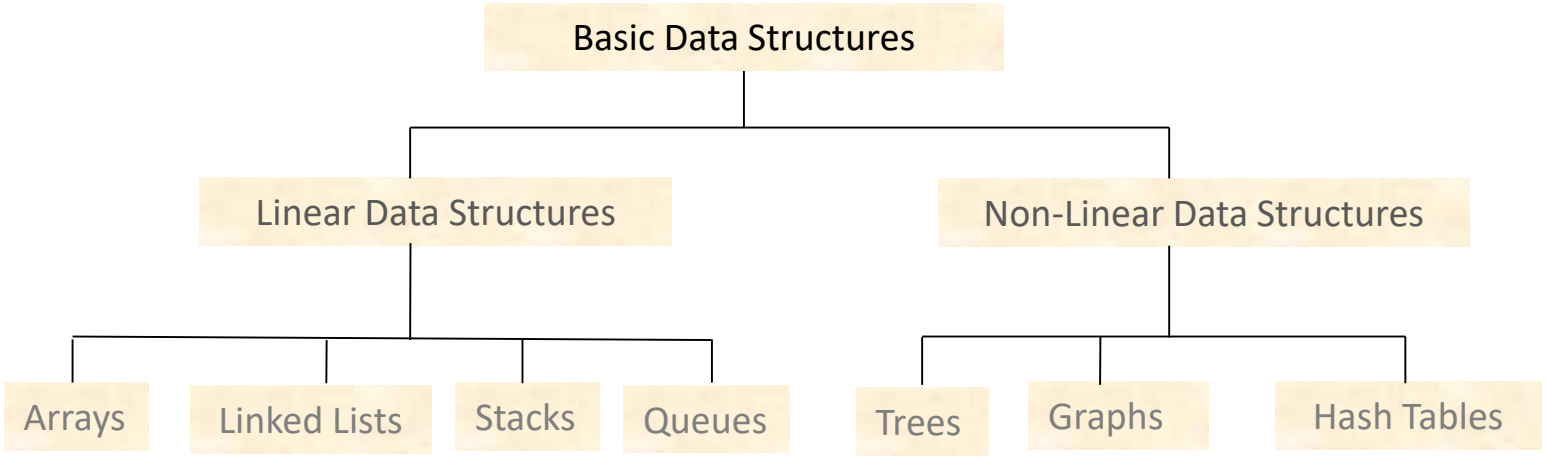
- To solve the above-mentioned problems, data structures is used.

KEY USES OF DATA STRUCTURES

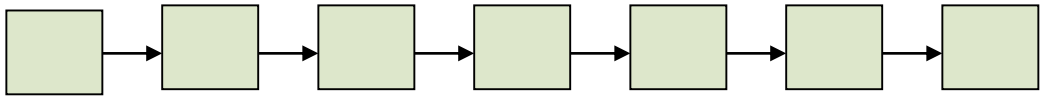
- Data structures find application in a wide range of fields:

- **Databases:** Trees and Hash tables Data structures are used for indexing and fast data retrieval in databases.
- **Operating Systems:** Queues and Stacks Data structures manage processes, memory allocation, and file systems.
- **Network Protocols:** Data structures like linked lists and trees are used to route data packets efficiently across networks.
- **Artificial Intelligence:** Data structures play a main role in representing knowledge, storing data for machine learning models, and implementing algorithms like search and optimization.

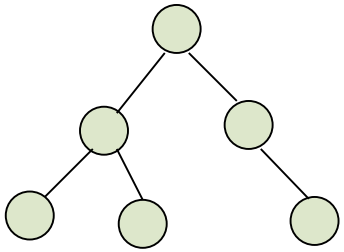
BASIC DATA STRUCTURE



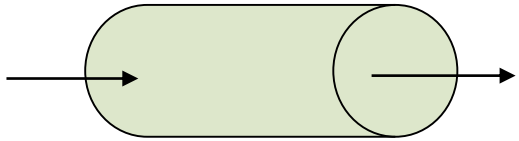
array



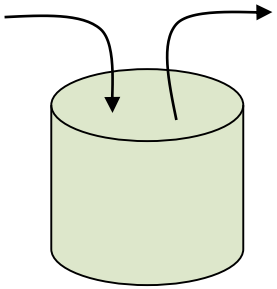
Linked list



tree



queue



stack

TYPES OF DATA STRUCTURE

- Linear : In Linear data structure, values are arrange in linear fashion.
  - Array: Fixed-size
  - Linked-list : Variable-size
  - Stack : Add to top and remove from top
  - Queue : Add to back and remove from front
  - Priority queue : Add anywhere, remove the highest priority

TYPES OF DATA STRUCTURE

- Non-Linear : The data values in this structure are not arranged in order.
  - Tree: Data is organized in branches.
  - Graph: A more general branching structure, with less strict connection conditions than for a tree.

ABSTRACT DATA TYPE (ADT)

- Abstract data type : an ADT is composed of a collection of data and a set of operations on that data.

- From this definition we can say that an *ADT is the interface of a data structure without any specification* of the *implementation.*

  - **Specifications of an ADT**
    - Indicate what the ADT operations do, not how to implement them.
  - **Implementation of an ADT**
    - Includes choosing a particular data structure.

    - *Data Structures represents the “Physical implementation of an ADT”

APPLICATIONS OF DATA STRUCTURES & ALGORITHMS (DSA)

- Following are some important categories of Data Structure algorithms:
  - Search – Algorithm to search an item in a data structure.
  - Sort – Algorithm to sort items in a certain order.
  - Insert – Algorithm to insert item in a data structure.
  - Update – Algorithm to update an existing item in a data structure.
  - Delete – Algorithm to delete an existing item from a data structure.

SELECTION OF DATA STRUCTURE

- The choice of particular data model *depends* on some consideration:
  - The size of the data.
  - The size of the storage.
  - The data dynamics, such as changing or editing the data.
  - The speed of data use
  - The structure should be simple enough that one can effectively process the data when necessary

SELECTING A DATA STRUCTURE

- Selecting a data structure to solve a problem includes the following three steps:

1. Analyze your problem to determine the basic operations that must be supported.

- *Examples:* basic operations include *inserting/deleting* a data item into/ from the data structure, and *finding* a specified data item.

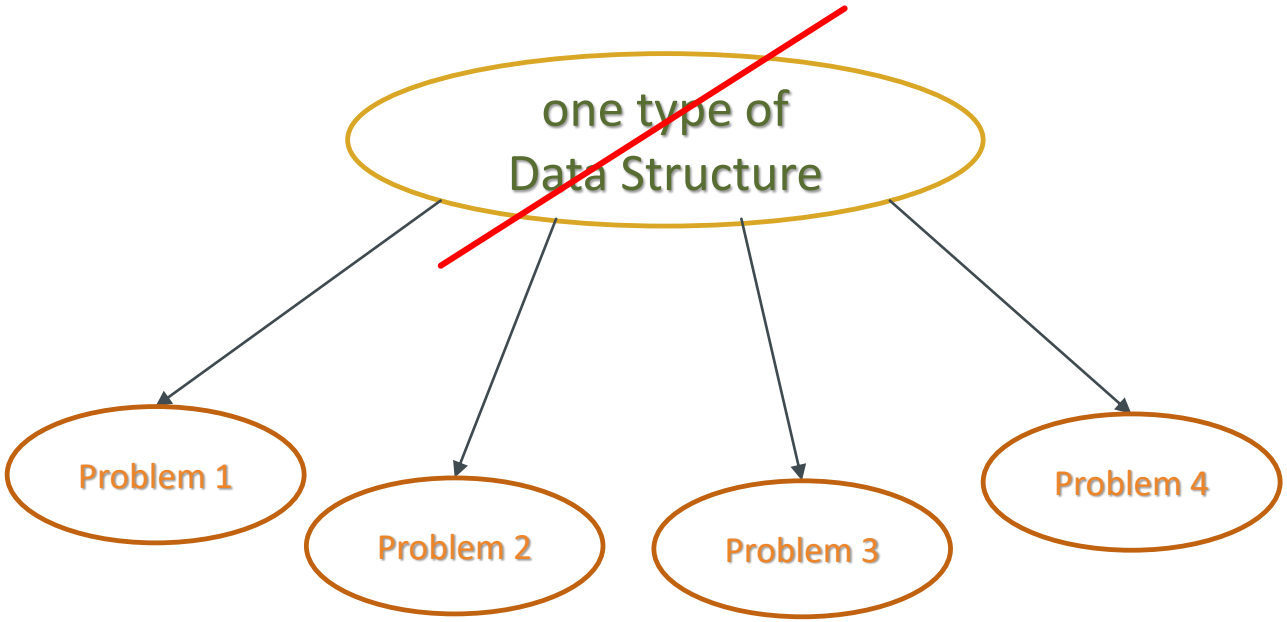
2. Quantitative analysis for each operation of the structure.

- *Examples:* Determining total *space* available to store the data and the *time* required to process the structure.

3. Choose the data structure that best meets these requirements.

CONCLUSION

- No single data structure works well for all purposes, and so it is important to know the strengths and limitations of several of them.



ALGORITHM

- An algorithm is a step by step procedure to solve a particular function.

- To solve any complex real life problems :

  - First, we have to define the problems.
  - Second, is to design the algorithm to solve that problem.

- There are few steps of refinement involved when a problem is converted to a program this method is called ***stepwise refinement method***.

STEPWISE REFINEMENT TECHNIQUE

- \- An appropriate mathematical model required to write an algorithm for a problem is called: *Stepwise Refinement Technique*.

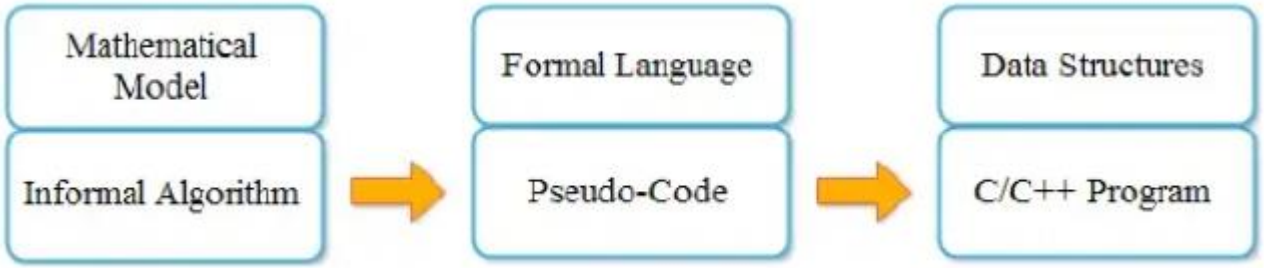
- **1.** In the **first stage**, modeling " Write an informal algorithm " : we try to represent the problem using an appropriate mathematical model such as a graph, tree etc.

- **2.** At the **next stage**, algorithm is written in pseudo-language. Convert algorithm to a formal one by applying any programming language syntax including less formal English statements.

- **3.** In the **final stage** Converting the formal algorithm by a programing language manual (C/C++ code).

STEPWISE REFINEMENT TECHNIQUE

- The three steps in refinement process.



ANALYSIS OF ALGORITHM

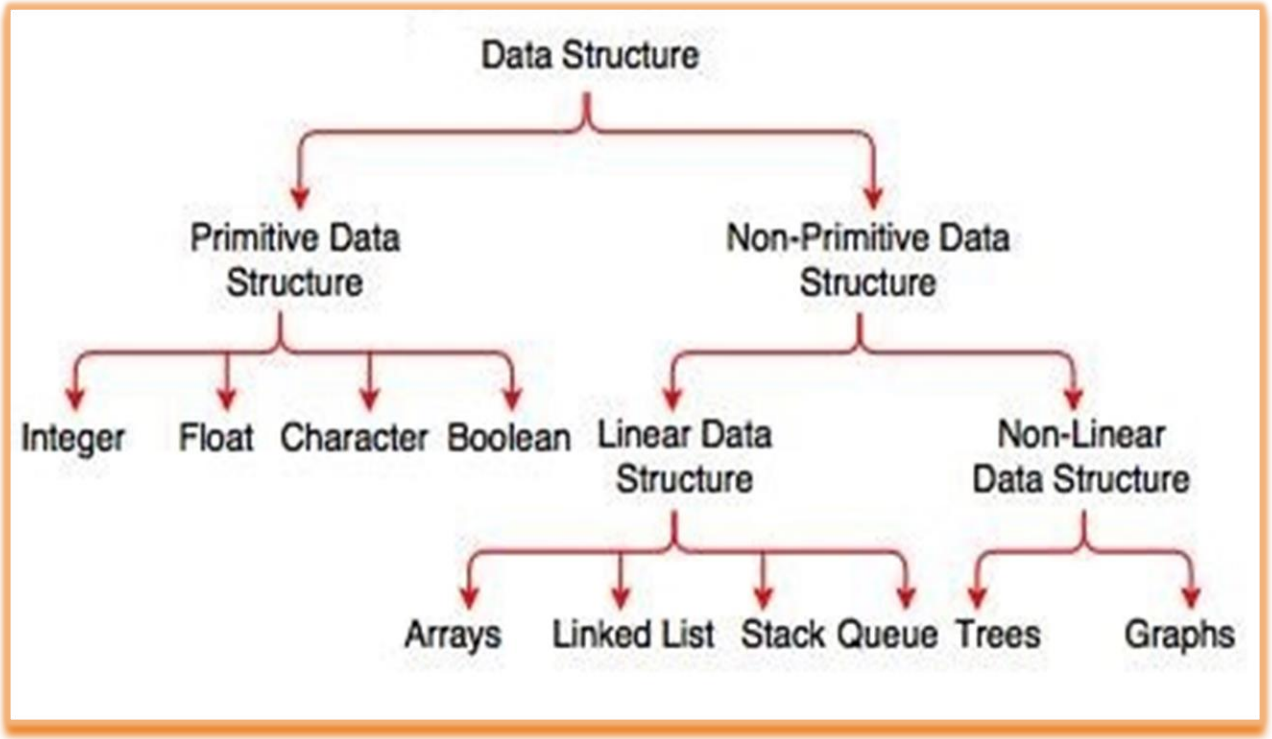
- An **algorithm** is correct analyzed if under all valid inputs it produces the correct output.

- **Efficiency (Complexity) Analysis:**

- We want the algorithm to make best use of

  - Space **->>** (storage) .
  - Time **->>** (how long it take to run, number of instructions executed, …etc.)

CLASSIFICATION OF DATA STRUCTURE



CLASSIFICATION OF DATA STRUCTURE

- **1.** Primitive data structures : A primitive data structure used to represent the basic data structures and are directly operated by the machine instructions, which is in a primitive level.
- All the basic constants (integers, floating numbers, character and string constants)and pointers are considered as primary data structures.

- **2.** Non-primitive data structures : It is data structure of a group of homogeneous (same type) or Non-homogeneous (different type) data items.
  - Homogenous: In this type of data structures, values of the same types of data are stored.
  - Array

  - Non-Homogenous: In this type of data structures, data values of different types are grouped and stored.
  - Structures
  - Classes