Data Structures And Algorithms

Lecture 1

ICBT Southern Campus

WHAT IS DATA?

- Data
  - A collection of row facts which can be processed to have conclusions
    e.g. Data:Sunil,25,Galle;

Types of data

- Textual: For example, your name (Amal)
- Numeric: For example, your student ID
- Audio: For example, your voice
- Video: For example, your voice and picture

DEFINITIONS:

- **ALGORITHM:**
  - Step-by-step procedure(set of instructions) which can be applied to data to solve a problem or achieve some goal. Eg: I'm thirsty...I want to make a tea

- **PROGRAM:**
  - Implements an algorithm.

- **DATA STRUCTURE:**
  - The manner in which data is represented in the computer to facilitate its access and manipulation by an algorithm.
    Or
    - **Organization** of data needed to solve the problem

DATA STRUCTURE

- A particular way of storing and organizing data in a computer so that it can be used efficiently and effectively.
- Data structure is the logical or mathematical model of a particular organization of data.
- A group of data elements grouped together under one name.
  - For example, an array of integers

Basic Terminology: Elementary Data Organization:

- Data: Data are simply values or sets of values. (any row thing)
  eg: 101, Nimal, A

- Data items: Data items refers to a single unit of values.
  eg: Nimal

- Data items that are divided into sub-items are called Group items.

Ex: An Employee Name may be divided into three subitems- first name, middle name, and last name.

- Data items that are not able to divide into sub-items are called Elementary items. Ex: SSN

Basic Terminology: Elementary Data Organization: Cont.

- Entity: An entity is something that has certain attributes or properties which may be assigned values

Eg: Employee-> Attributes- ID, Name, Age, Gender, Address
Values- E001, Nayani, 34, F, Colombo

Basic Terminology: Elementary Data Organization: Cont.

- Range – Lower and the upper limit/range of values of an attribute

Eg: Age: 18-25

- Information – Processed data

Eg: ID: 101

Name : Nayani

Basic Terminology: Elementary Data Organization: Cont.

- Field - It is a single elementary unit of information representing an attribute of an entity. (Columns)

Eg: ID

- Record - It is the collection of field values of a given entity. (Considering rows)

Eg: E001,Nayani,35,F,Colombo

- File- It is the collection of records of the entities in a given entity set

- Primary key – Unique key which is used to retrieve a specific record

Basic Terminology: Elementary Data Organization: Cont.

- Records may also be classified according to length. A file can have fixed-length records or variable-length records.
  - In fixed-length records, all the records contain the same data items with the same amount of space assigned to each data item. Eg: mobile number – 10 digits
  - In variable-length records file records may contain different lengths. Eg: Name of an employee
- Database – A collection of files
- DBMS – A platform to manage databases

Classification of Data Structures Cont.

Data structures are generally classified into

- Primitive data Structures
- Non-primitive data Structures

Primitive data Structures: Primitive data structures are the fundamental data types which are supported by a programming language.(Can be directly manipulated by machine instructions)
Eg: In Java - >int,byte,short,long,float,double,Boolean,char

Non- Primitive data Structures: Non-primitive data structures are those data structures which are created using primitive data structures.They are not in atomic in nature.
Eg: Arrays, stacks, linked lists, Queues, trees, and graphs.

Classification of Data Structures Cont.

Based on the structure and arrangement of data, non-primitive data structures is further classified into

1. Linear Data Structure
2. Non-linear Data Structure

1. Linear Data Structure: A data structure is said to be linear if its elements form a sequence or a linear list. There are basically two ways of representing such linear structure in memory.

   1) One way is to have the linear relationships between the elements represented by means of sequential memory location. These linear structures are called arrays.

   2) The other way is to have the linear relationship between the elements represented by means of pointers,references or links. These linear structures are called linked lists.

   Eg: Arrays, Queues, Stacks, Linked lists

Classification of Data Structures Cont.

2. Non-linear Data Structure: A data structure is said to be non-linear if the data are not arranged in sequence or a linear. This structure is mainly used to represent data containing a hierarchical relationship between elements.

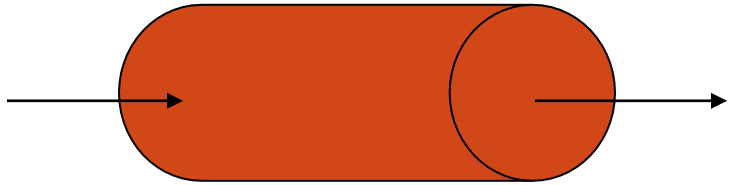
Eg: Trees and graphs

Data Structure	**Advantages**	**Disadvantages**
Array	Quick insertion, very fast access if index known.	Slow search, slow deletion, fixed size.
Ordered array	Quicker search than unsorted array.	Slow insertion and deletion, fixed size.
Stack	Provides last-in, first-out access.	Slow access to other items.
Queue	Provides first-in, first-out access.	Slow access to other items.
Linked List	Quick insertion, quick deletion.	Slow search.
Binary Tree	Quick search, insertion, deletion (if tree remains balanced)	Deletion algorithm is complex.

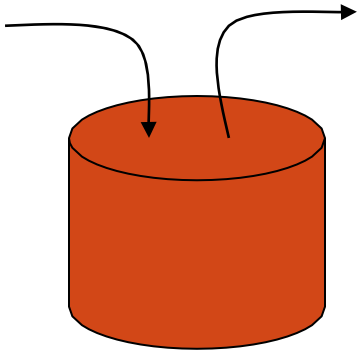
Types of data structures

Array

Linked List

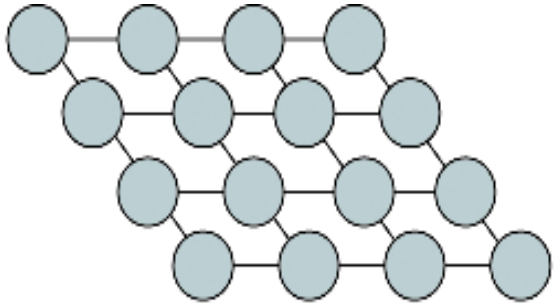


Queue

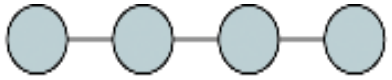


Stack

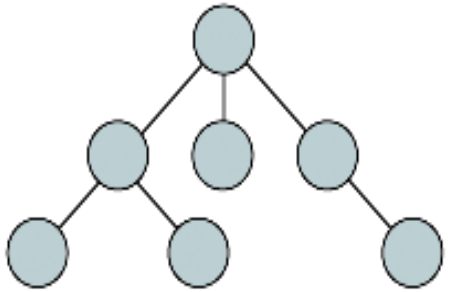
Types of data structures



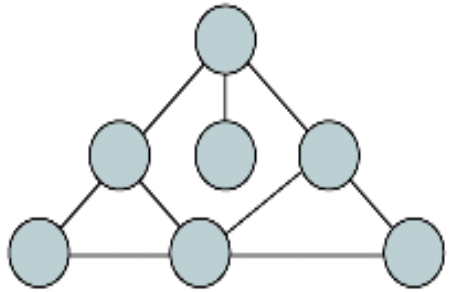
(a) Matrix



(b) Linear list



(c) Tree



(d) Graph

Data Structures Operations

The data appearing in data structures are processed by means of certain operations. They are,

1. Traversing: accessing each record/node exactly once so that certain items in the record may be processed. (This accessing and processing is sometimes called “visiting” the record.)

2. Searching: Finding the location of the desired node with a given key value, or finding the locations of all such nodes which satisfy one or more conditions.

3. Inserting: Adding a new node/record to the structure.

4. Deleting: Removing a node/record from the structure.

Data Structures Operations Cont.

The following two operations, which are used in special situations:

1. Sorting: Arranging the records in some logical order
(e.g., alphabetically according to some NAME key, or in numerical order according to some NUMBER key, such as social security number or account number)

2. Merging: Combining the records in two or more data structures to create a single unified data structure
(e.g. Merging two arrays and creating a new array)

THE NEED FOR 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 main motivation to learn and understand data structures)

Real world usage of Data Structures

- Real-world data storage.
  - To keep the details of a set of people.
- Programmer's tools
  - The data structures that are meant for the access of the program itself.
    - Stacks, queues,...
- Real-world modelling
  - To model real world situations.
    - graphs, queues,...

WHY WE NEED DATA STRUCTURES?

- Allow us to achieve an important object-oriented programming goal: component reuse.

- Once each data structure has been implemented, it can be used over and over again in various applications.

- Data structure is a particular way of storing and organizing information in a computer, so that it can be retrieved and used most productively.

WHY STUDY DATA STRUCTURES AND ALGORITHMS?

Programs are comprised of two things:

data and **algorithms. …**

The reason for learning about **data structures** is because adding **structure** to our **data** can make the algorithms much simpler, easier to maintain, and often faster.

END