جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني

Department of Cyber Security

Subject: Data Structure

Class: Second

Lecturer: Asst. Prof. Dr. Ali Kadhum Al-Quraby

Lecture: (1)

Introduction to Data Structures

Study Year: 2024-2025

Introduction to Data Structures

Data Structure is an arrangement of data in a computer's memory (or sometimes on a disk). Data structures include arrays, linked lists, stacks, binary trees, and hash tables, among others. Algorithms manipulate the data in these structures in various ways, such as searching for a particular data item and sorting the data.

What is Data Structure?

- A scheme for organizing related pieces of information
- A way in which sets of data are organized in a particular system
- An organized aggregate of data items
- A computer interpretable format used for storing, accessing, transferring and archiving data
- The way data is organized to ensure efficient processing: this may be in lists, arrays, stacks, queues or trees

Overview of Data Structures

Another way to look at data structures is to focus on their strengths and weaknesses. Table 1.1 shows the advantages and disadvantages of the various data structures.

TABLE 1.1 Characteristics of Data Structures

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.
Red-black tree	Quick search, insertion, deletion. Tree always balanced.	Complex.
2-3-4 tree	Quick search, insertion, deletion. Tree always balanced. Similar trees good for disk storage.	Complex.
Hash	tableVery fast access if key known. Fast insertion	Slow deletion, access slow if key not known, inefficient memory usage.
Heap	Fast insertion, deletion, access to largest item.	Slow access to other items.
Graph	Models real-world situations.	Some algorithms are slow and complex.

The data structures shown in Table 1.1, except the arrays, can be thought of as Abstract Data Types, or ADTs.

A data type consists of

- a domain (= a set of values)
- A set of operations.

Example 1: *Boolean* or *logical* data type provided by most programming languages.

- Two values: true, false.
- Many operations, including: AND, OR, NOT, etc.

Abstract Data Type (**ADT**): The basic idea behind an abstract data type is the **separation of the use of the data type from its implementation** (i.e., what an abstract data type does can be specified separately from how it is done through its implementation)

The advantages of using the **ADT** approach are:

1. The implementation of **ADT** can change without affecting those method that use the **ADT**
2. The complexity of the implementation are hidden

For example, an **abstract stack data structure** could be defined by two operations: **push**, that inserts some data item into the structure, **and pop**, that extracts an item from it.

Advantages of data structures

The major advantages of data structures are:
• It gives different level of organizing data.
• It tells how data can be stored and accessed in its elementary level

Operation on Data Structures

The operations that can be performed on data structures are:
Creation: This operation creates a data structure. The declaration statement causes space to be created for data upon entering at execution time.

Destroy: This operation destroys the data structure and aids in the efficient use of memory.

Selection: This operation is used to access data within a data structure. The form of selection depends on the type of data structure being accessed.

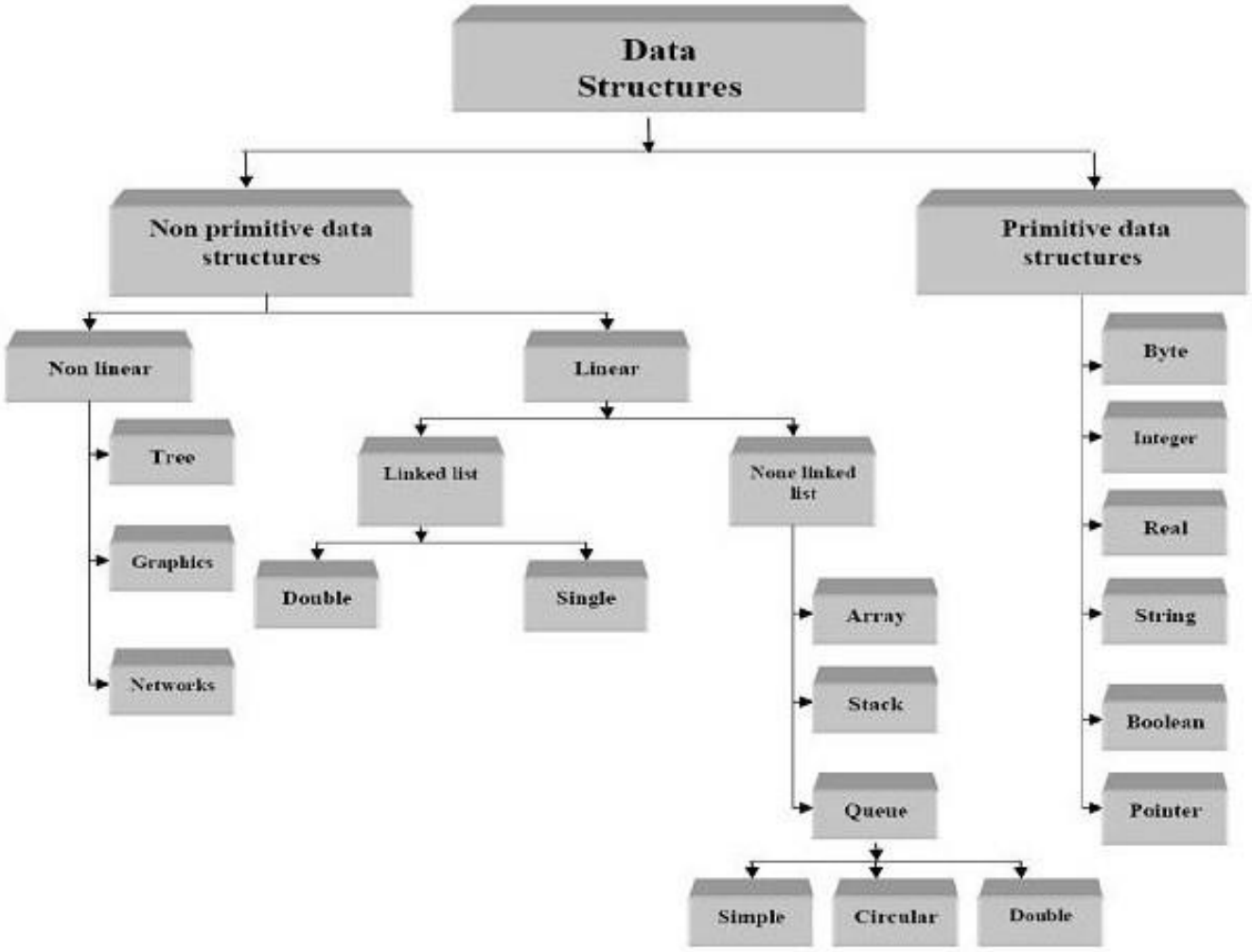
Update: This operation changes or modifies the data in the data structure and it is an important property in selection operation.

Types of Data Structures

Linear Data Structure: Stacks, Queues, Linked Lists, etc.

Non-linear Data Structure: Trees, Graphs, etc .

As illustrated in block diagram below.



Block diagram of Data Structures

Difference between Linear and Nonlinear Data Structures

Main difference between linear and nonlinear data structures lie in the way they organize data elements. In linear data structures, data elements are organized sequentially and therefore they are easy to implement in the computer's memory. In nonlinear data structures, a data element can be attached to several other data

elements to represent specific relationships that exist among them. Non-Linear data structure is that if one element can be connected to more than two adjacent element then it is known as non-linear data structure.

Student Feedback

Here are some questions and multiple-choice options for your lecture on **"Introduction to Data Structures":**

Question 1: What is a data structure?

A. A scheme for organizing related pieces of information.
B. A way in which sets of data are organized in a particular system.
C. An organized aggregate of data items.
D. A computer interpretable format used for storing, accessing, transferring, and archiving data.
E. The way data is organized to ensure efficient processing: this may be in lists, arrays, stacks, queues, or trees.

Answer: E. The way data is organized to ensure efficient processing: this may be in lists, arrays, stacks, queues, or trees.

Question 2: Which of the following data structures provides first-in, first-out access?

A. Array
B. Stack
C. Queue
D. Linked list
E. Binary tree

Answer: C. Queue

Question 3: What is an Abstract Data Type (ADT)?

A. A data type that is not well-defined.
B. A type of data structure.
C. A data type consisting of only integers.
D. The way data is organized in memory.
E. The separation of the use of a data type from its implementation.

Answer: E. The separation of the use of a data type from its implementation.

Question 4: What is the main advantage of using an Abstract Data Type (ADT)?

A. It simplifies the programming language.

B. It makes data structures more complex.
C. The implementation of ADT can change without affecting methods that use the ADT.
D. It eliminates the need for data structures.
E. It always results in faster algorithms.

Answer: C. The implementation of ADT can change without affecting those methods that use the ADT.

Question 5: Which of the following is NOT a type of data structure mentioned in the lecture?

A. Array
B. Graph
C. Set
D. Linked list
E. Hash table

Answer: C. Set

Question 6: What operation creates a data structure and allocates space for data?

A. Destroy
B. Selection
C. Update
D. Creation
E. Deletion

Answer: D. Creation

Question 7: What is the main difference between linear and nonlinear data structures?

A. Linear data structures are always faster.
B. Nonlinear data structures use less memory.
C. Linear data structures organize data sequentially, while nonlinear data structures represent specific relationships among data elements.
D. Nonlinear data structures are used exclusively in programming languages.
E. Linear data structures are more complex than nonlinear ones.

Answer: C. Linear data structures organize data sequentially, while nonlinear data structures represent specific relationships among data elements.

Question 8: Which of the following is a non-linear data structure?

A. Array
B. Stack
C. Queue

D. Linked list
E. Graph

Answer: E. Graph

Question 9: What is the advantage of using data structures?

A. It simplifies code.
B. It reduces memory usage.
C. It tells how data can be stored and accessed.
D. It eliminates the need for algorithms.
E. It makes data manipulation slower.

Answer: C. It tells how data can be stored and accessed.

Question 10: What type of access does a stack provide?

A. First-in, first-out access.
B. Random access.
C. Last-in, first-out access.
D. No access.
E. Quick access to other items.

Answer: C. Last-in, first-out access.