Data Structures

Md. Jalal Uddin

Lecturer

Department of CSE

City University

Email: jalalruice@gmail.com

No: 01717011128 (Emergency Call)

?

✓ Data are simple values or set of values.

✓ Data item refers to a single unit of values

✓ Data items that are divided into sub-items are called group items.

For example:

An employee’s name may be divided into three sub item…..

❖ First name

❖Middle name

❖Last Name.

But, NID number would be normally be treated as a single item

Data Type and Data Structure

❑ **Data type**

- Set of possible values for variables
- Operations on those values

Ex : int, float, char ……….

❑ **Data Structure**

➢ A data structure is an arrangement of data in a computer's memory or even disk storage.

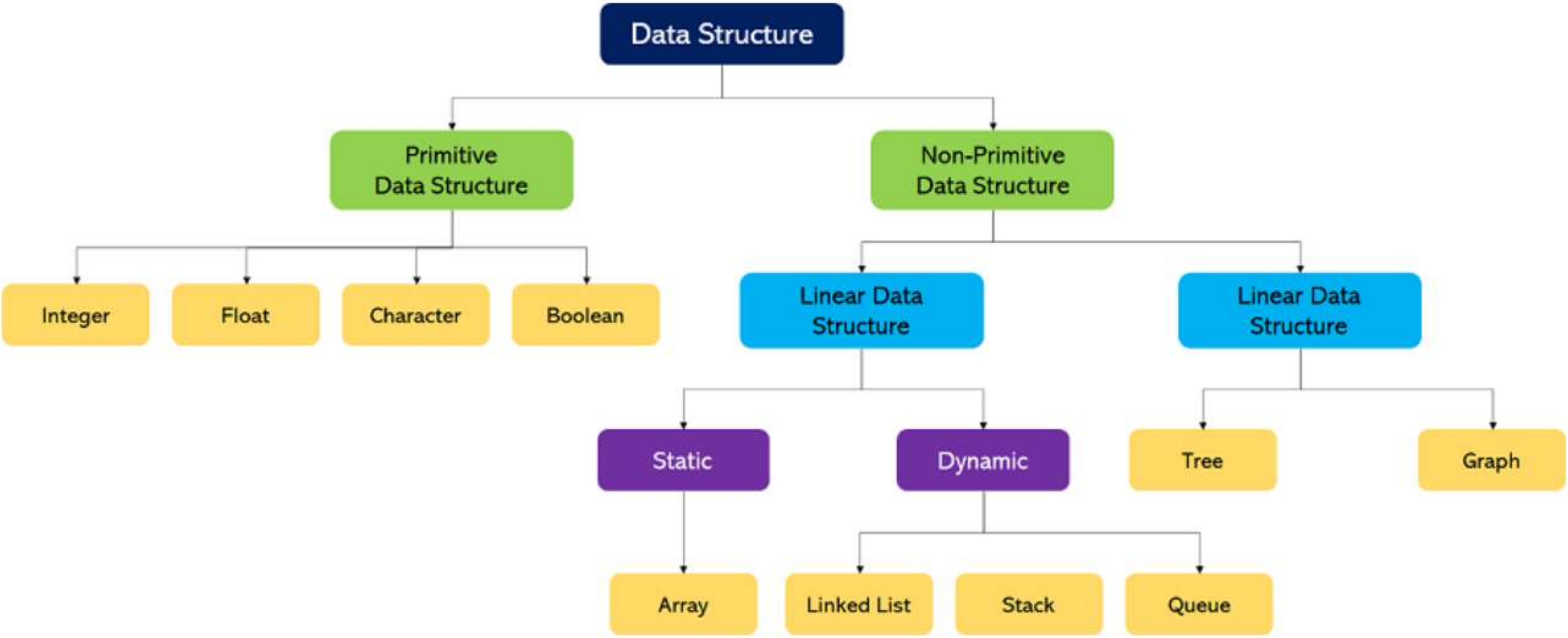
➢ The logical and mathematical model of a particular organization of data is called a data structure.

➢ A **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently.

❑ **Examples of several common data structures:**

- arrays,
- linked lists,
- stacks,
- queues,
- trees,
- and Graph

Classification of Data Structures



Classification of Data Structures Cont…

1. Primitive Data structure

The primitive data structures are primitive data types. The int, char, float, double, and pointer are the primitive data structures that can hold a single value.

- **Primitive Data Structures** are the data structures consisting of the numbers and characters that come **in-built** into programs.
- These data structures can be manipulated or operated directly by machine-level instructions.
- Basic data types like **Integer, Float, Character** and **Boolean** come under the Primitive Data Structures. These data types are also called **Simple data types**, as they contain characters that can't be divided further**.**

Classification of Data Structures Cont…

2. Non-Primitive Data structure

- **Non-Primitive Data Structures** are those data structures derived from Primitive Data Structures.
- These data structures can't be manipulated or operated directly by machine-level instructions.
- The focus of these data structures is on forming a set of data elements that is either **homogeneous** (same data type) or **heterogeneous** (different data types).

The non-primitive data structure is divided into two types:

- Linear data structure
- Non-linear data structure

Classification of Data Structures Cont…

Linear Data Structures:

- A data structure that preserves a linear connection among its data elements is known as a Linear Data Structure.
- The arrangement of the data is done linearly, where each element consists of the successors and predecessors except the first and the last data element.

Based on memory allocation, the Linear Data Structures are further classified into two types:

Static Data Structures: The data structures having a fixed size are known as Static Data Structures. The memory for these data structures is allocated at the compiler time, and their size cannot be changed by the user after being compiled

Example: **Array**

Dynamic Data Structures: The data structures having a dynamic size are known as Dynamic Data Structures. The memory of these data structures is allocated at the run time, and their size varies during the run time of the code. Example: **Linked Lists, Stacks**, and **Queues.**

Classification of Data Structures Cont…

Non-Linear Data Structures:

- Data structures where data elements are not arranged sequentially or linearly are called **non-linear data structures**.
- In a non-linear data structure, single level is not involved. Therefore, we can’t traverse all the elements in single run only.
- Non-linear data structures are not easy to implement in comparison to linear data structure.
- It utilizes computer memory efficiently in comparison to a linear data structure.
- Its examples are trees and graphs.

Types of Data Structure

There are two types of data structure

Linear Data Structure

A data structure is said to be linear if its elements form a sequence, or in other words a linear list.

- Array
- Stack
- Queue
- Linked List

Non- Linear Data Structure

A non-linear structure is mainly used to represent data containing a hierarchical relationship between elements.

- Tree
- Graph

Algorithms

- An algorithm is a well-defined list of step for solving problem.
- An Algorithm is a finite step – by – step list of well defined instructions for solving a particular problem.
- It is used to manipulate the data contained in the data structures as in searching and sorting.

Algorithms Cont…

Characteristics of an algorithm:

- ✓ Unambiguous: Algorithm should be clear and Unambiguous.
- ✓ Input: 0 or more well defined input.
- ✓ Output: 1 or more well defined output.
- ✓ Finiteness: Algorithm must be terminated after a finite number of steps.
- ✓ Feasibility: Feasible with the available resource.
- ✓ Independent: Independent of any programming language.

Algorithms Cont…

The following are the steps required to add two numbers entered by the user:

Step 1: Start

Step 2: Declare three variables a, b, and sum.

Step 3: Enter the values of a and b.

Step 4: Add the values of a and b and store the result in the sum variable, i.e., sum=a+b.

Step 5: Print sum

Step 6: Stop

Complexity

- The complexity of an algorithm is the function which gives the running time and/or space in terms of the input size.
- The complexity of an algorithm is the function f(n) which measures the running time and/or space in terms of the input size n

There are two main complexity measures of the efficiency of an algorithm:

- ✓ **Time complexity** is a function describing the amount of time an algorithm takes in terms of the amount of input to the algorithm.

- ✓ ***Space complexity*** is a function describing the amount of memory (space) an algorithm takes in terms of the amount of input to the algorithm.

  -Data space

  -Environment space

  - Instruction space

Complexity Cont…

Three Cases are usually investigate in complexity :

1. Worst case : The maximum value of f(n) for any possible input.
2. Average case : The expected value of f(n) for any possible input.
3. Best case : The minimum value of f(n) for any possible input.

Asymptotic Notation

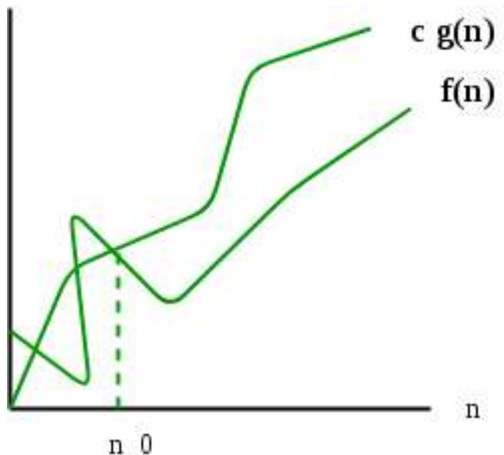
The commonly used asymptotic notations used for calculating the running time complexity of an algorithm is given below:

- Big oh Notation (?)
- Omega Notation (Ω)
- Theta Notation (θ)

Asymptotic Notation Cont…

Big oh Notation (?)

- ✓ Big-O notation represents the upper bound of the running time of an algorithm.
- ✓ It returns the highest possible output value (big-O)for a given input.
- ✓ Therefore, it gives the worst-case complexity of an algorithm.

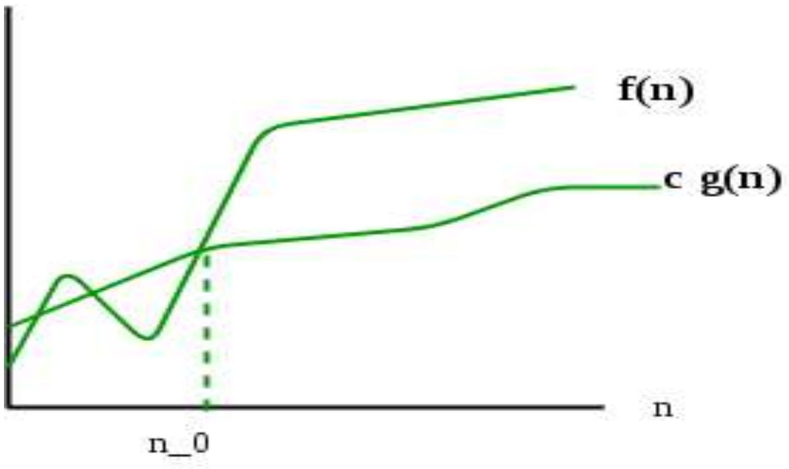


f(n) = O(g(n))

Asymptotic Notation Cont…

Omega Notation (Ω)

- ✓ Omega notation represents the lower bound of the running time of an algorithm.
- ✓ Thus, it provides the best case complexity of an algorithm.

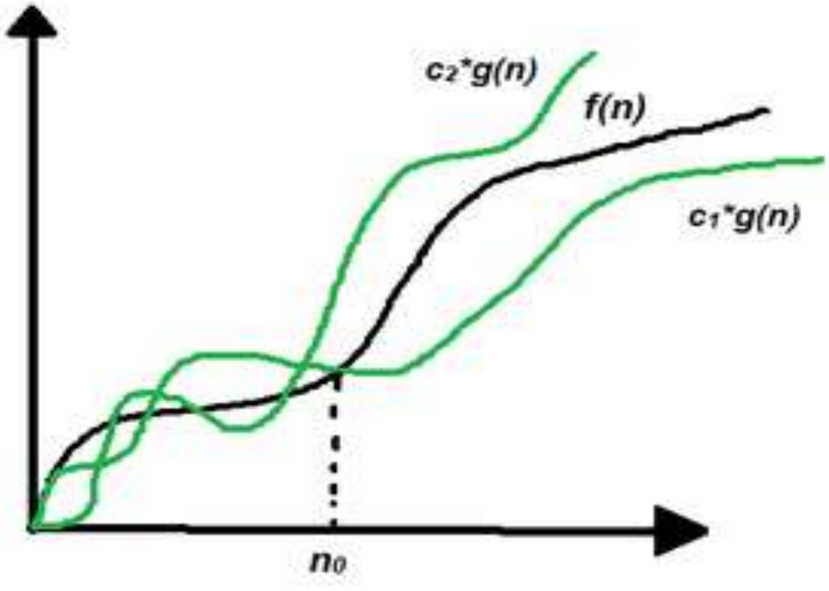


f(n) = Omega(g(n))

Asymptotic Notation Cont…

Theta Notation (θ)

- ✓ Theta notation encloses the function from above and below.
- ✓ Since it represents the upper and the lower bound of the running time of an algorithm.
- ✓ it is used for analyzing the **average-case** complexity of an algorithm.



Array

- **Linear array (One dimensional array) :** A list of finite number n of similar data elements referenced respectively by a set of n consecutive numbers, usually 1, 2, 3,…..n. That is a specific element is accessed by an index.

- Let, Array name is A then the elements of A is : a_1,a_2….. a_n
- Or by the bracket notation A[1], A[2], A[3],…………., A[n]
- The number k in A[k] is called a subscript and A[k] is called a subscripted variable.

Example

- A linear array STUDENT consisting of the name of six students

STUDENT

1	Dalia Rahaman
2	Sumona
3	Mubtasim Fuad
4	Anamul Haque
5	Ibtisam Rahaman
6	Jarin

Here, STUDENT[4] denote Anamul Haque

Array (con…)

➢ Linear arrays are called one dimensional arrays because each element in such an array is referenced by one subscript.

✓ **(Two dimensional array)** : Two dimensional array is a collection of similar data elements where each element is referenced by two subscripts.

Such arrays are called matrices in mathematics and tables in business applications.

✓ **Multidimensional arrays** are defined analogously

MATRICES

	1	2	3	4
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12
4	13	14	15	16

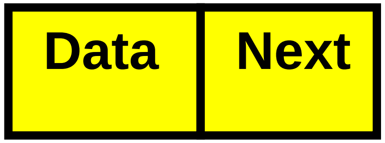
Here, MATRICES[3,3]=11

- **Array Data Structure**
  - It can hold multiple values of a single type.
  - Elements are referenced by the array name and an *ordinal* index.
  - Each element is a *value*
  - Indexing begins at zero.
  - The array forms a contiguous list in memory.
  - The name of the array holds the address of the first array element.
  - We specify the array size at compile time, often with a named constant.

Linked lists

- A linked list, or one way list, is a linear collection of data elements, called nodes, where the linear order is given by means of pointers.
- Dynamically allocate space for each element as needed.

Node

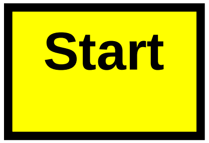


In linked list

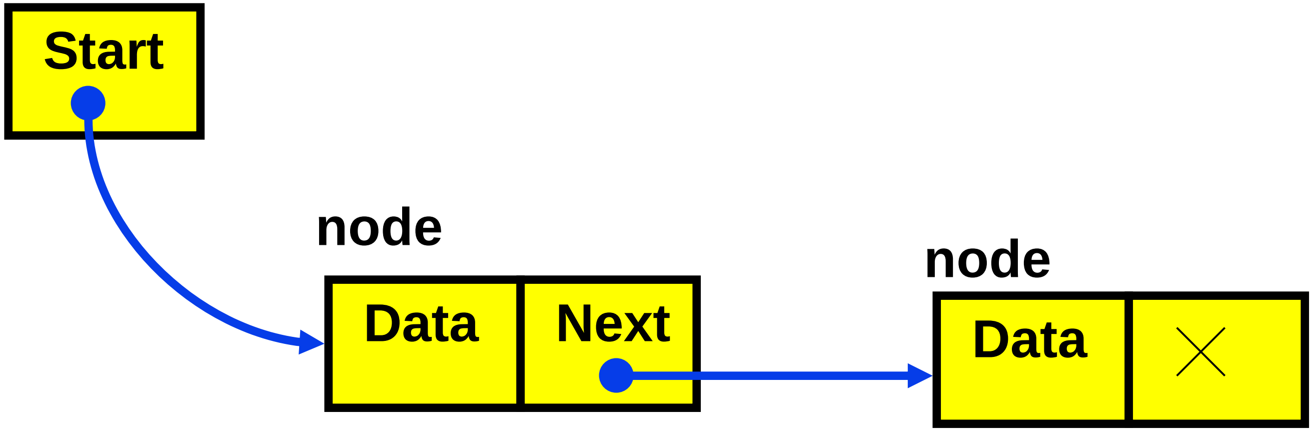
Each node of the list contains the data item
a pointer to the next node

Collection structure has a pointer to the list Start

Initially NULL

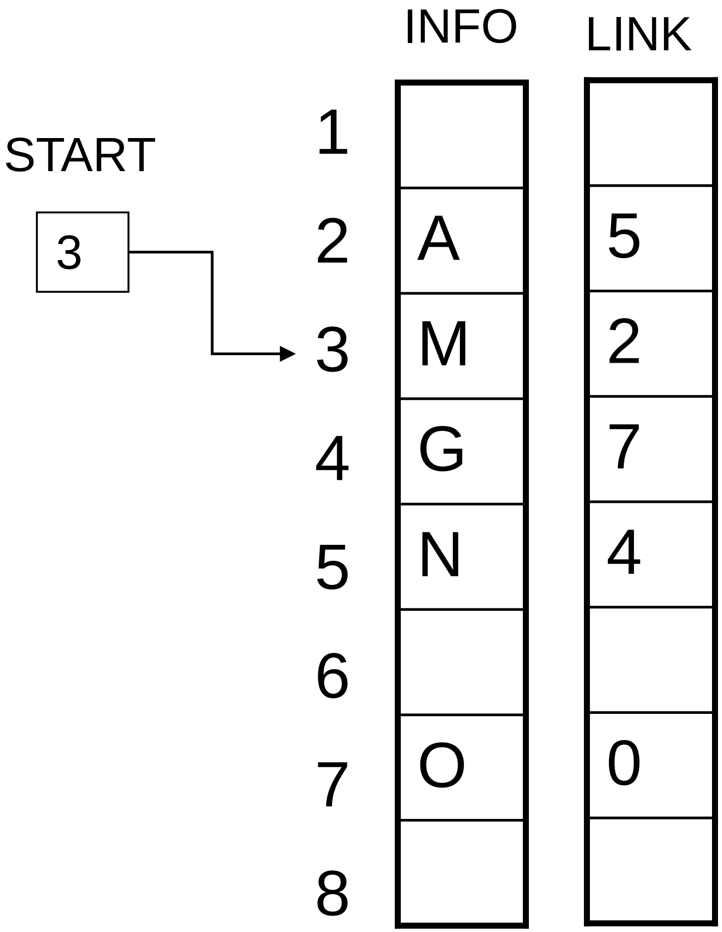


Linked lists Cont...



Linked list with 2 nodes

Linked lists



START

3

	INFO	LINK
1		
2	A	5
3	M	2
4	G	7
5	N	4
6		
7	O	0
8		

START=3, INFO[3]=M

LINK[3]=2, INFO[2]=A

LINK[2]=5, INFO[5]=N

LINK[5]=4, INFO[4]=G

LINK[4]=7, INFO[7]=O

LINK[7]=0, NULL value, So the list has ended

Data structure operations

The following four operations play a major role:

- ✓ **Traversing**
  Accessing each record exactly once so that certain items in the record may be processed. (This accessing or processing is sometimes called 'visiting" the records.)
- ✓ **Searching**
  Finding the location of the record with a given key value, or finding the locations of all records, which satisfy one or more conditions.
- ✓ **Inserting**
  Adding new records to the structure.
- ✓ **Deleting**
  Removing a record from the structure.

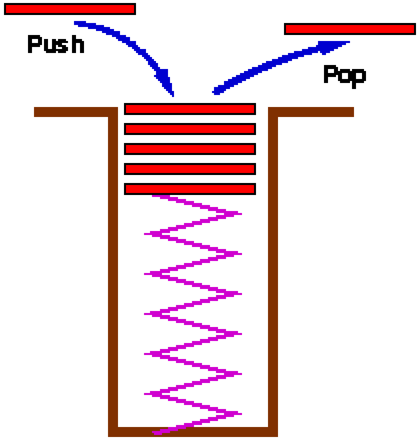
Data structure operations Cont…

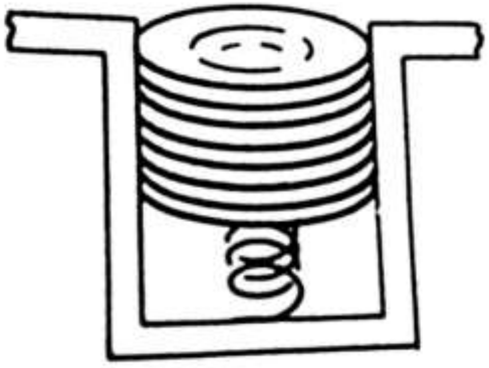
The following two operations, which are used in special situations, will also be considered:

- ✓ **Sorting:** Arranging the records in some logical order
- ✓ **Merging:** Combining the records in two different sorted files into a single sorted files

Stacks

- Stacks are a special form of collection with LIFO semantics
- Two methods
  - add item to the top of the stack
  - remove an item from the top of the stack
- Like a plate stacker

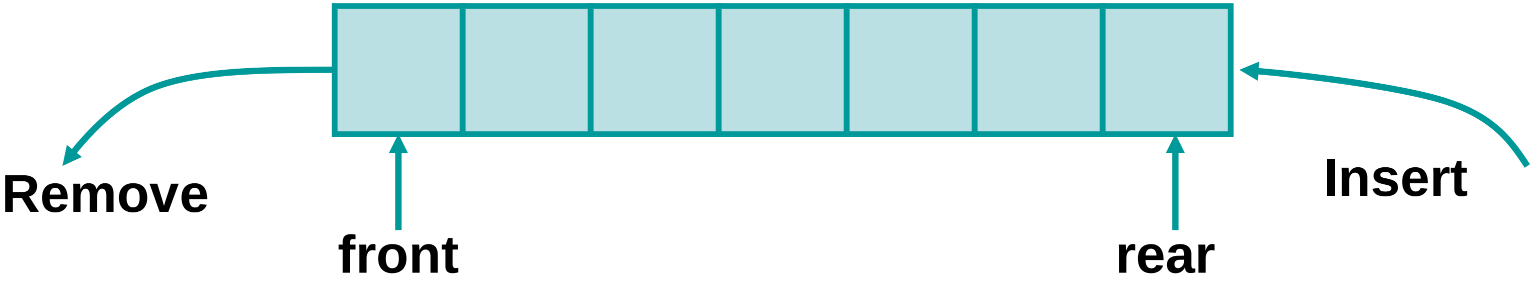




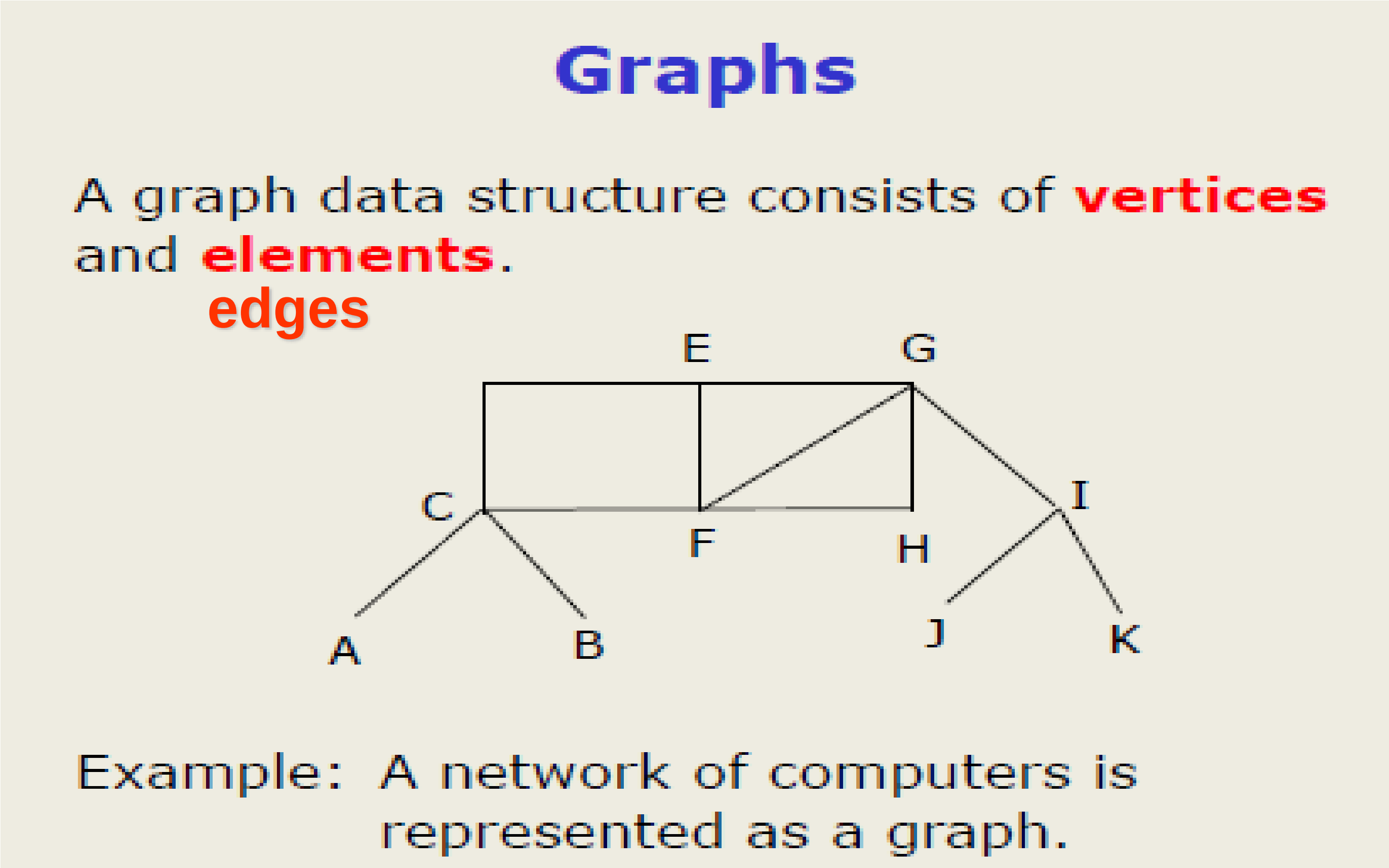
(a) Stack of dishes.

Queues

- Like a stack, a *queue* is also a list. However, with a queue, insertion is done at one end, while deletion is performed at the other end
- The insertion end is called *rear*
  - The deletion end is called *front*



(b) Queue waiting for a bus.

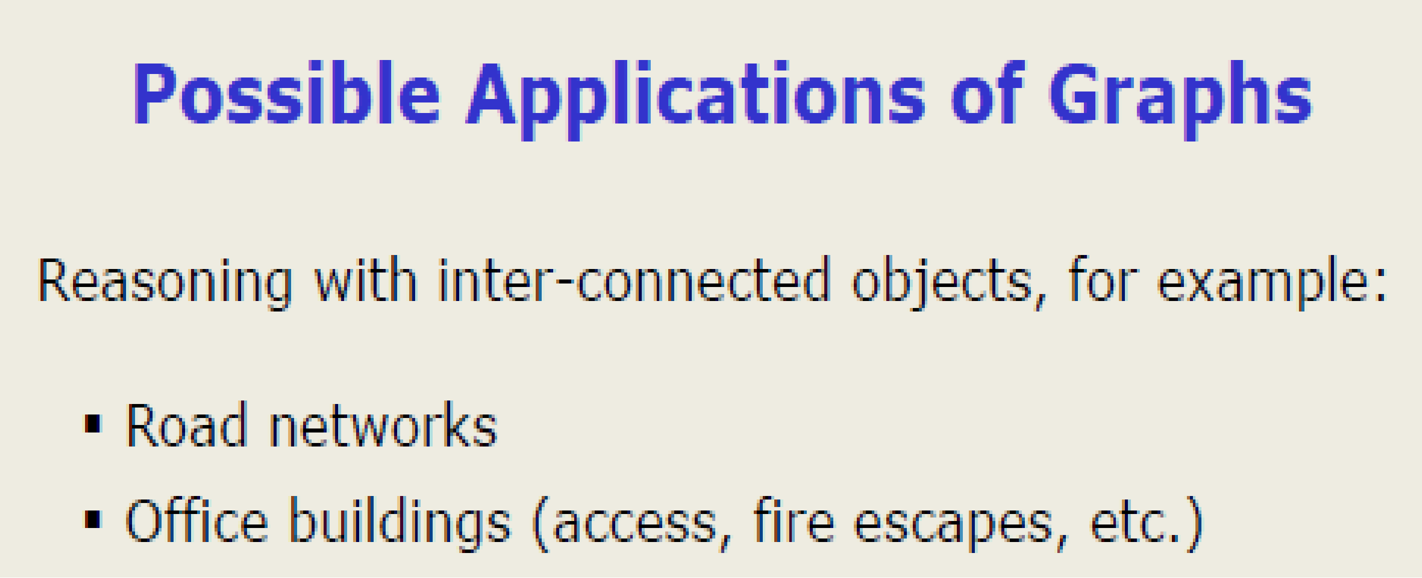


Graphs

A graph data structure consists of **vertices** and **elements**.

edges

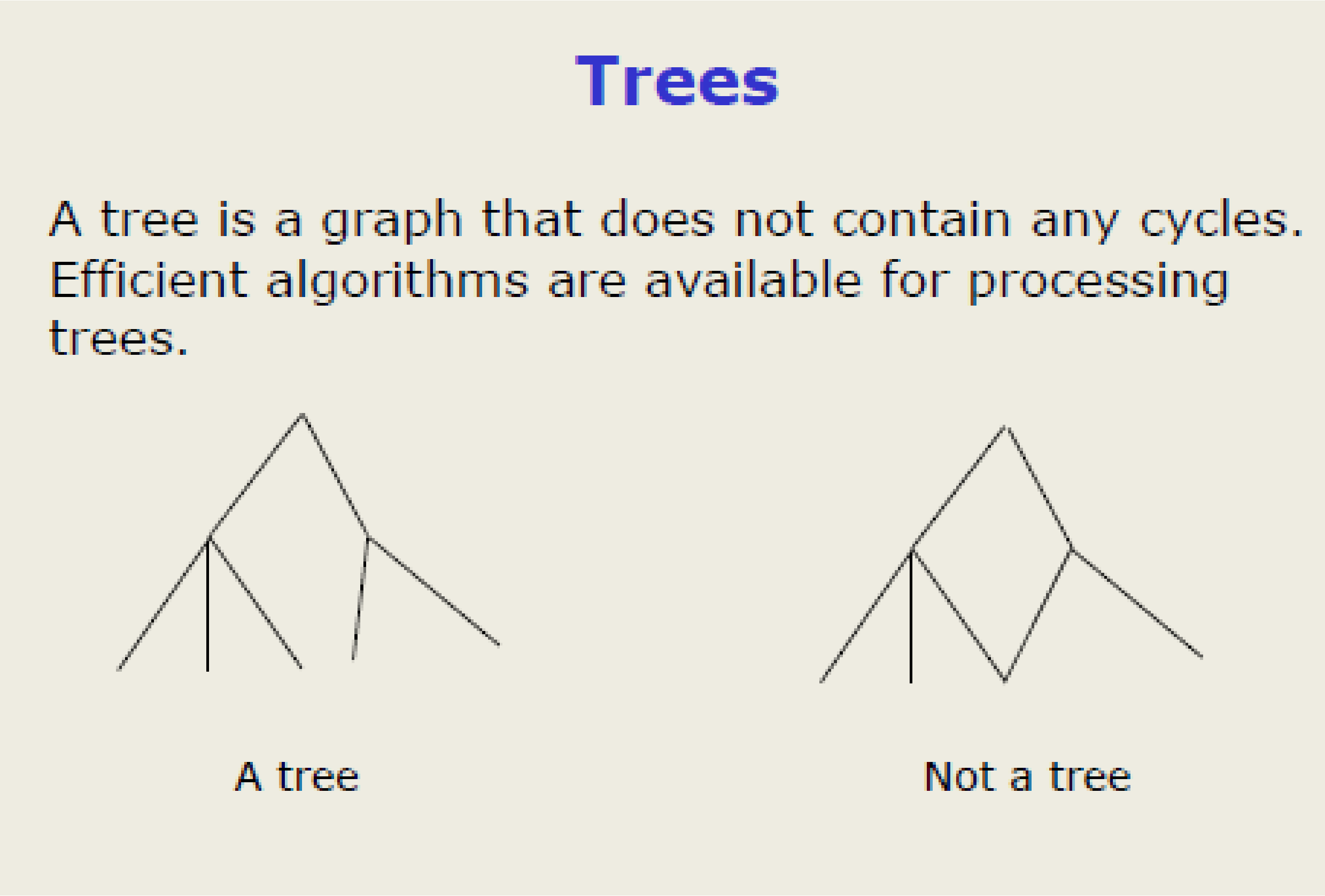
Example: A network of computers is represented as a graph.



Possible Applications of Graphs

Reasoning with inter-connected objects, for example:

- Road networks
- Office buildings (access, fire escapes, etc.)



Trees

A tree is a graph that does not contain any cycles. Efficient algorithms are available for processing trees.

A tree

Not a tree

Your Task