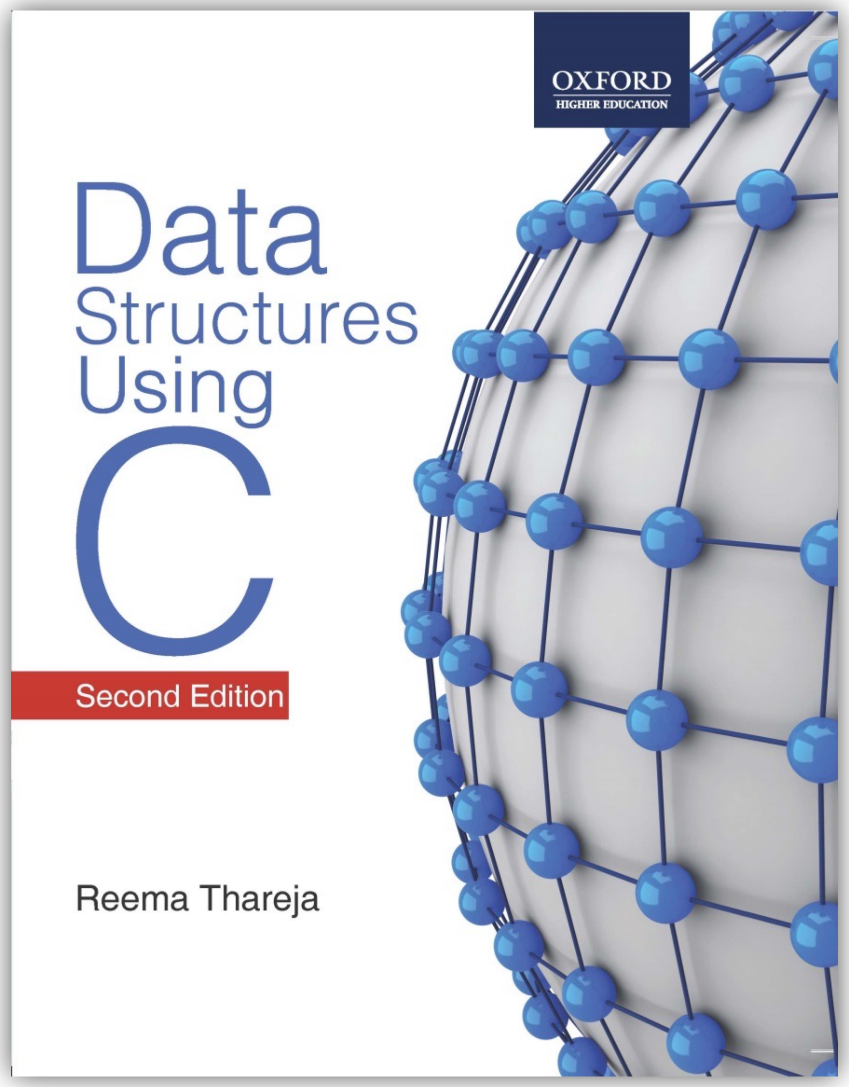
OXFORD
HIGHER EDUCATION



Data Structures Using C, 2e

Reema Thareja

Chapter 2

Introduction to Data Structures and Algorithms

Introduction

A *data structure* is an arrangement of data either in computer's memory or on the disk storage.

Some common examples of data structures are arrays, linked lists, queues, stacks, binary trees, graphs, and hash tables.

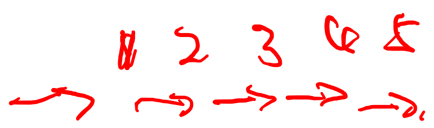
Data structures are widely applied in areas like:

- Artificial Intelligence
- Compiler design
- Operating system
- Statistical analysis package
- Database Management Systems (DBMS)
- Numerical analysis
- Simulation

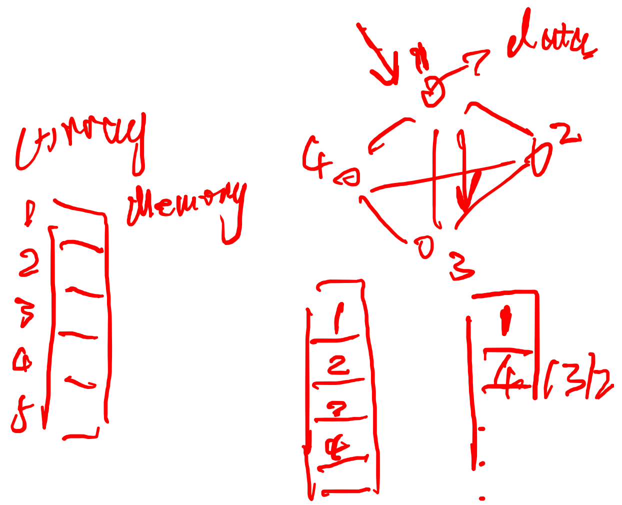
Classification of Data Structures

- Primitive data structures are the fundamental data types which are supported by a programming language. Some basic data types are integer, real, and boolean. The terms 'data type', 'basic data type', and 'primitive data type' are often used interchangeably. int arr[10];
- Non-primitive data structures are those data structures which are created using primitive data structures. Examples of such data structures include linked lists, stacks, trees, and graphs. Array?

✓Non-primitive data structures can further be classified into two categories: *linear* and *non-linear* data structures.



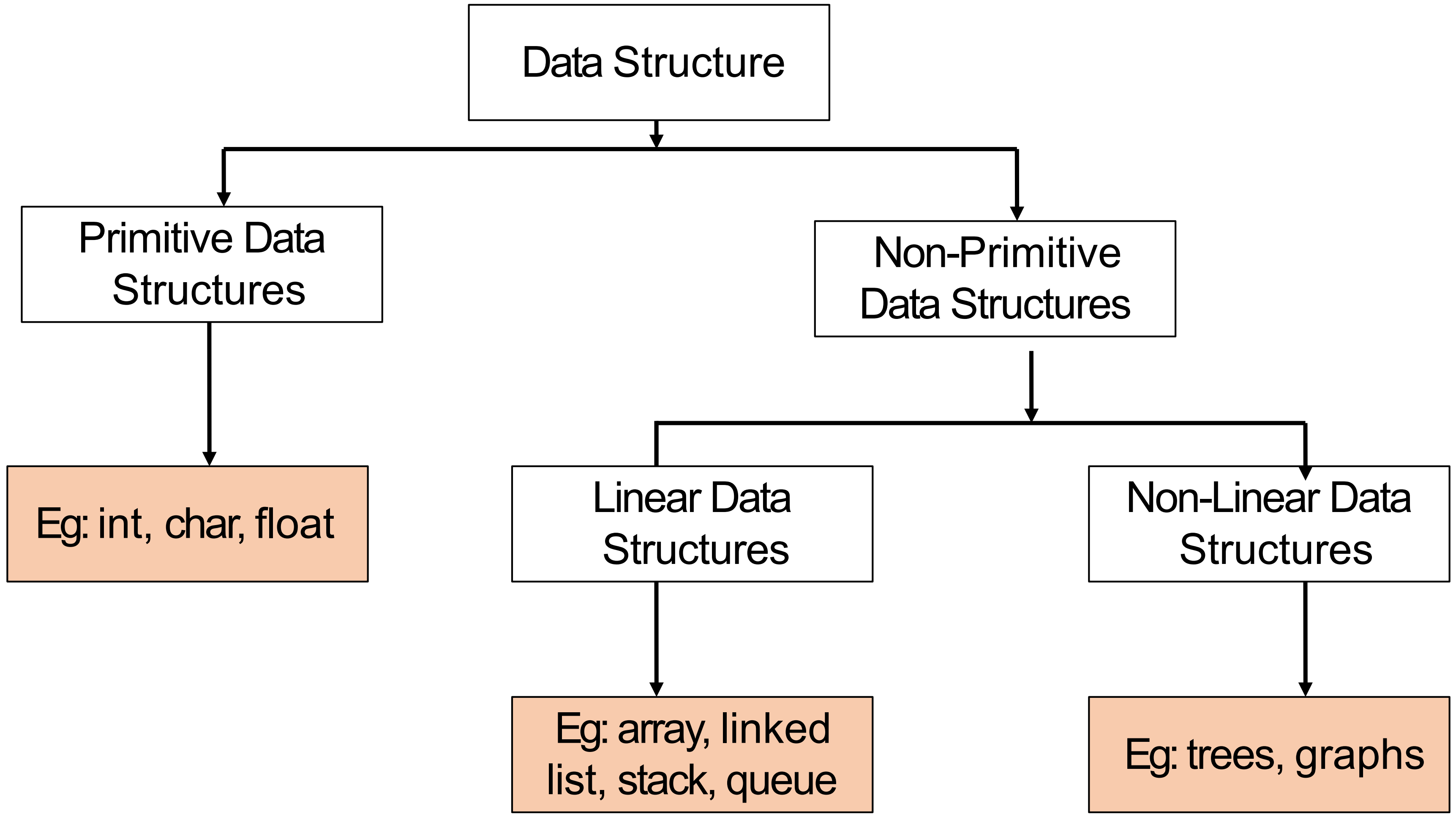
Classification of Data Structures



If the elements of a data structure are stored in a linear or sequential order, then it is a *linear* data structure. Examples are arrays, linked lists, stacks, and queues.

If the elements of a data structure are not stored in a sequential order, then it is a *non-linear* data structure. Examples are trees and graphs.

Classification of Data Structures

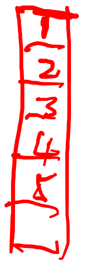

Data Structure
Primitive Data Structures
Non-Primitive Data Structures
Eg: int, char, float
Linear Data Structures
Non-Linear Data Structures
Eg: array, linked list, stack, queue
Eg: trees, graphs

Arrays

int
boolean
float

```
struct Node
{ int ID;
  String Name;
}
```

- An array is a collection of similar data elements.
- The elements of an array are stored in consecutive memory locations and are referenced by an index (also known as the subscript).

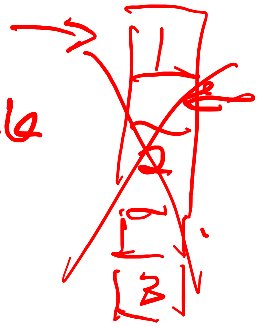


[8] * size of int

- Arrays are declared using the following syntax:

variable

type name[size];   eg. int arr[10];



1st element	2nd element	3rd element	4th element	5th element	6th element	7th element	8th element	9th element	10th element
int marks[0]	int marks[1]	marks[2]	marks[3]	marks[4]	marks[5]	marks[6]	marks[7]	marks[8]	marks[9]

Linked Lists

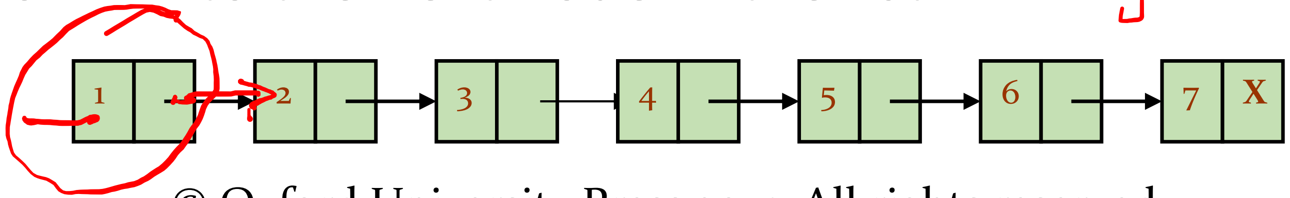
[illegible]

- A linked list is a very flexible dynamic data structure in which elements can be added to or deleted from anywhere.
- In a linked list, each element (called a *node*) is allocated space as it is added to the list.
- Every node in the list points to the next node in the list. Therefore, in a linked list every node contains two types of information:

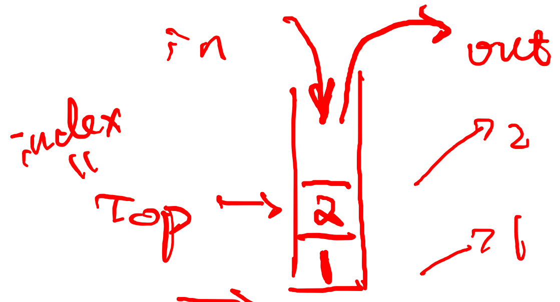
✓ The data stored in the node

✓ A pointer or link to the next node in the list

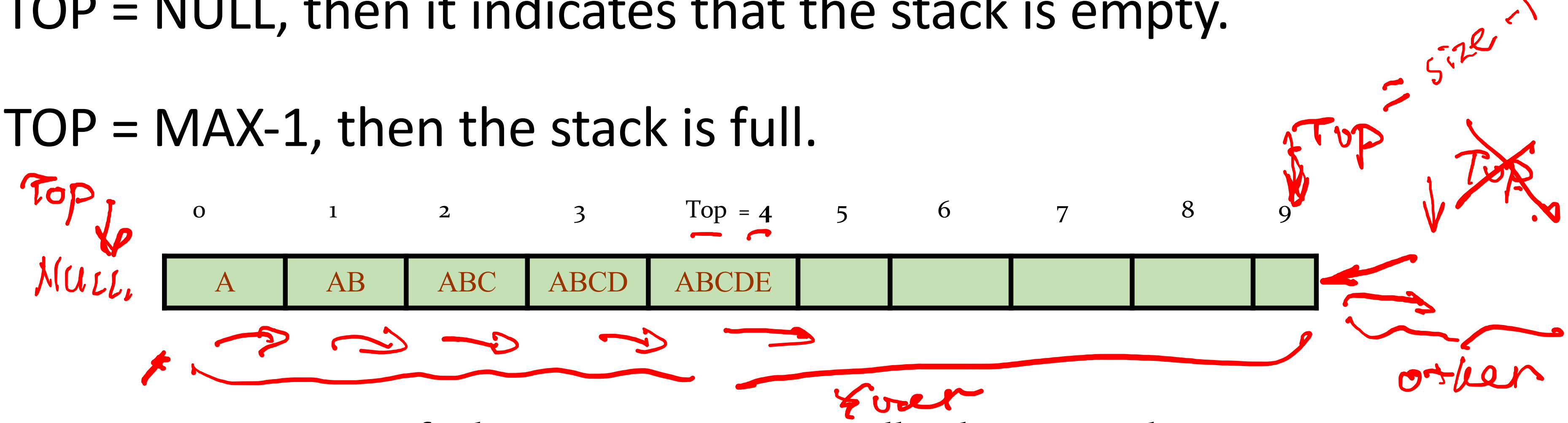
```
struct Node {
    int data;
    struct Node* next;
}
```



Stack

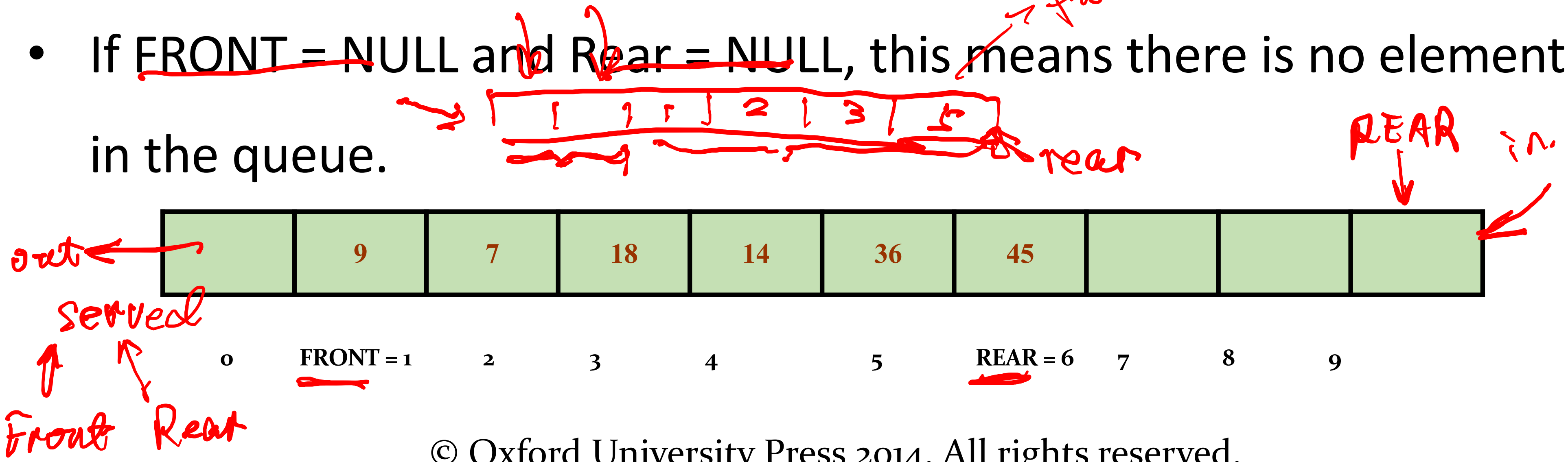


- A stack is a last-in, first-out (LIFO) data structure in which insertion and deletion of elements are done only at one end, known as TOP of the stack.
- Every stack has a variable TOP associated with it, which is used to store the address of the topmost element of the stack.
- If TOP = NULL, $\neq 0$ then it indicates that the stack is empty.
- If TOP = MAX-1, then the stack is full.



Queue

- A queue is a FIFO (first-in, first-out) data structure in which the element that is inserted first is the first one to be taken out.
- The elements in a queue are added at one end called the REAR and removed from the other one end called FRONT.
- When REAR = MAX – 1, then the queue is full. full?,
- If FRONT = NULL and Rear = NULL, this means there is no element in the queue.



Lab Submission Instructions - Revision

- **Upload Files**
  - **Video**: Upload the recorded video to OneDrive.
  - **Solution Code**: Upload the solution code to your GitHub repository.
- **File Size Limits on GitHub**
  - Files added via the browser: **Up to 25 MiB per file.**
  - Larger files (up to **100 MiB**): Use the **command line** to upload.
  - Learn more: GitHub File Upload Guide
- **Submit Links**
  - Submit both the **video link** and the **code link** on **Blackboard**.