UNIVERSITY OF COMPUTER STUDIES (PATHEIN)

Stack and Queue

Faculty of Computer Science

Learning Objectives

- To gain the knowledge about the stack and queue
- To understand how to allocate stack and queue data structures on memory
- To discuss the operations of the stack and queue

Stack

What is a Stack ?

- Stack is also called last in first out **(LIFO)** or First in last out **(FILO)**system
- Stack is a linear data structure which follows a particular order in which the operations are performed.
- Insertion of element into stack is called **PUSH** and deletion of element from stack is called **POP**.

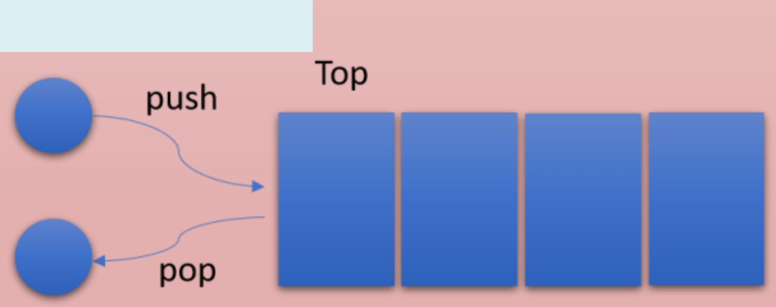


Figure : Push, Pop, Top of Stack

Memory Management

- The stack can be implemented into two ways:
  - Using arrays (Static implementation)
  - Using pointer (Dynamic implementation)

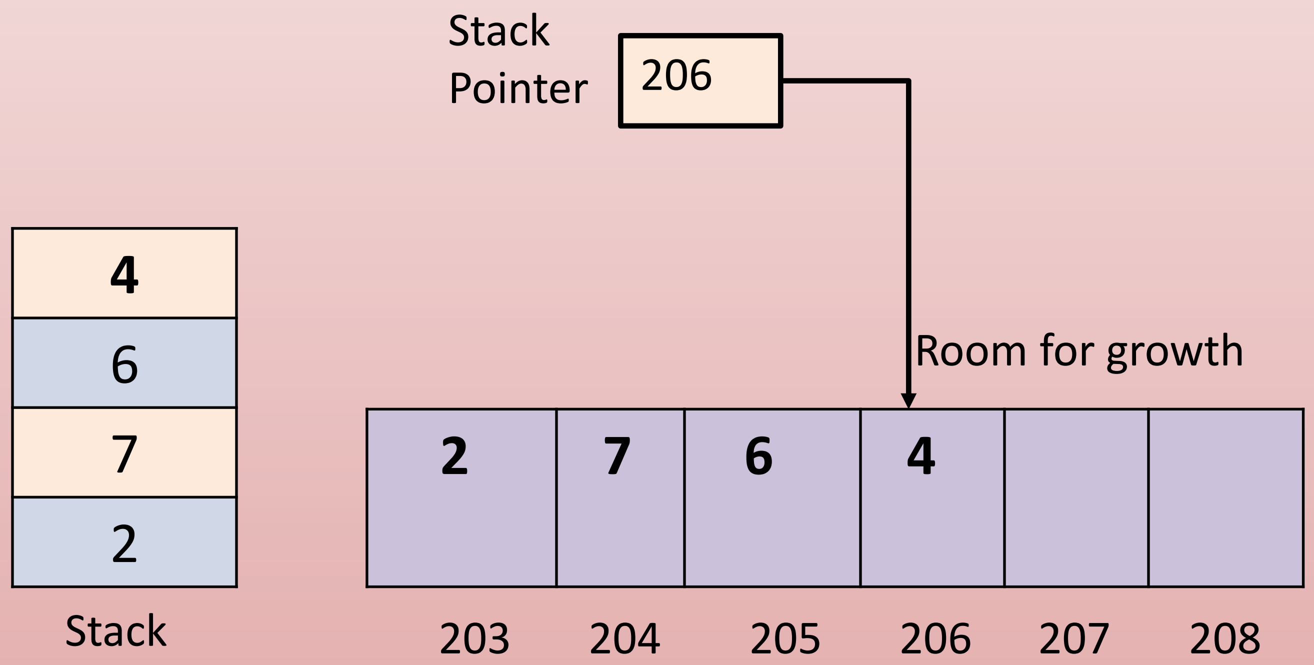


Figure: Stack Pointers

Operation on Stacks

- Stack( ): It creates a new stack that is empty. It needs no parameter and returns an empty stack.
- push(item): It adds a new item to the top of the stack.
- pop( ): It removes the top item from the stack.
- peek( ): It returns the top item from the stack but does not remove it.
- isEmpty( ): It tests whether the stack is empty.
- size( ): It returns the number of items on the stack.

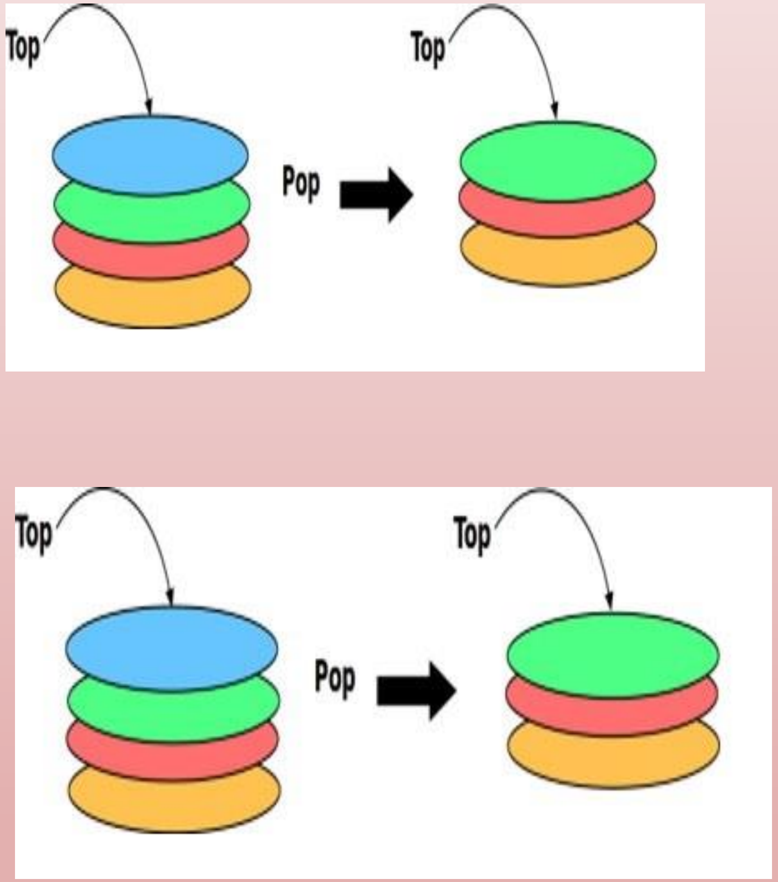


Figure: Push and Pop operations in Stack

Stack Conditions

- Depending on implementation may be necessary to check if stack is full, attempt to add item to a full stack is an overflow error.

- Important to know if stack is empty, attempt to remove an item from an empty stack is an underflow error.

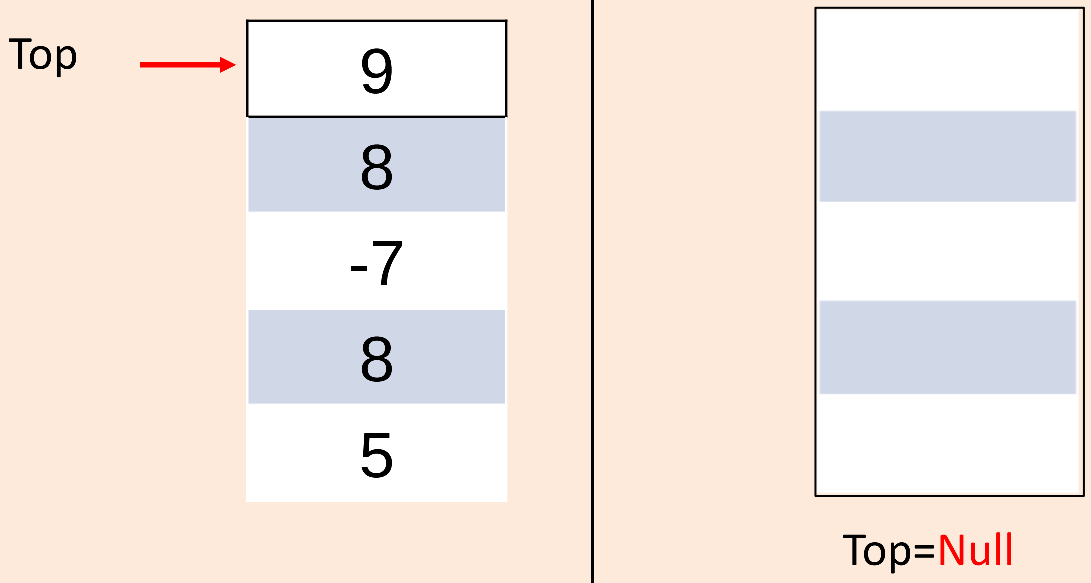


Figure : Overflow and Underflow Error

PUSH Operation

- The process of adding one element or item to the stack is represented by an operation called as the PUSH operation.

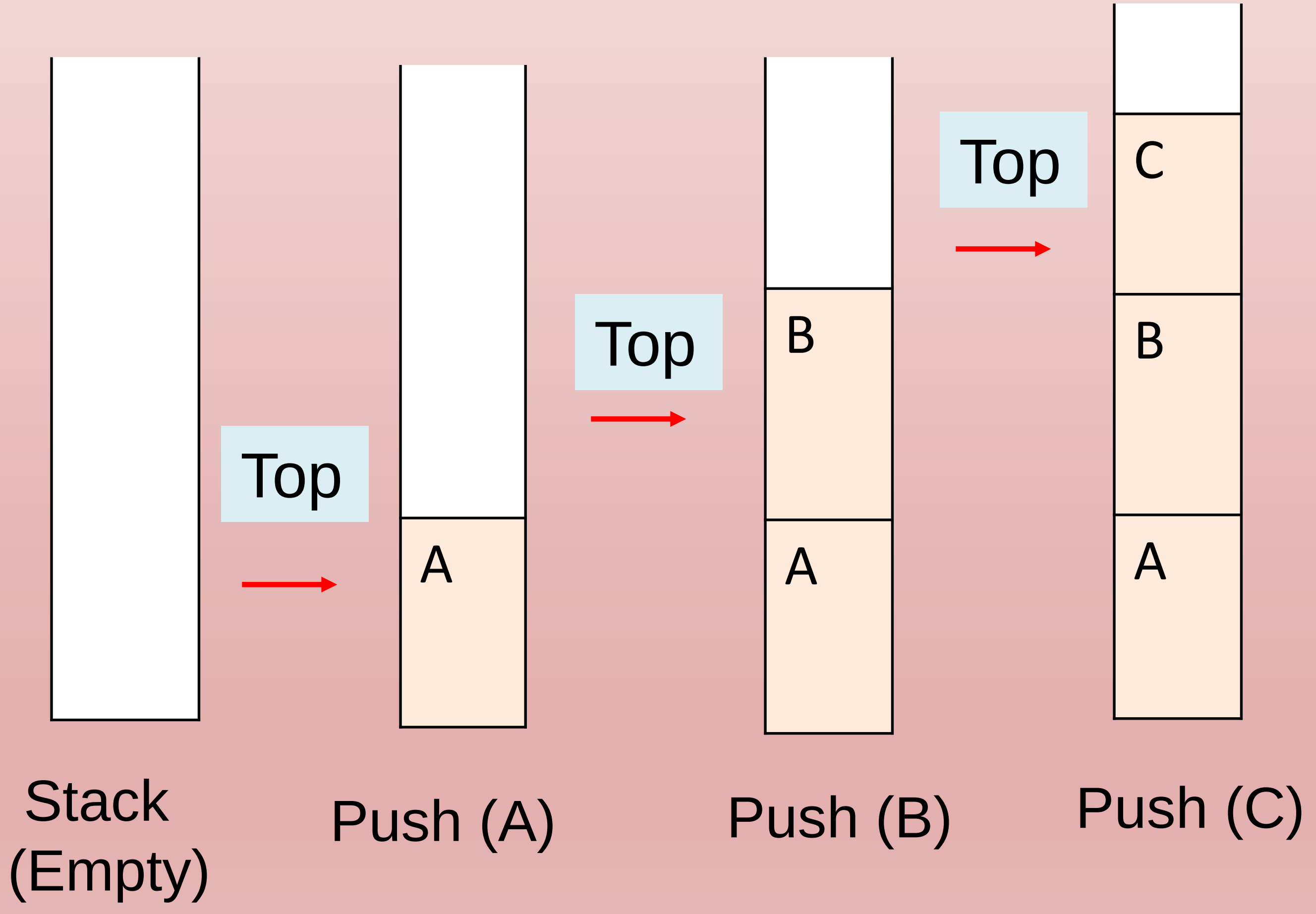


Figure : Push operation in Stack

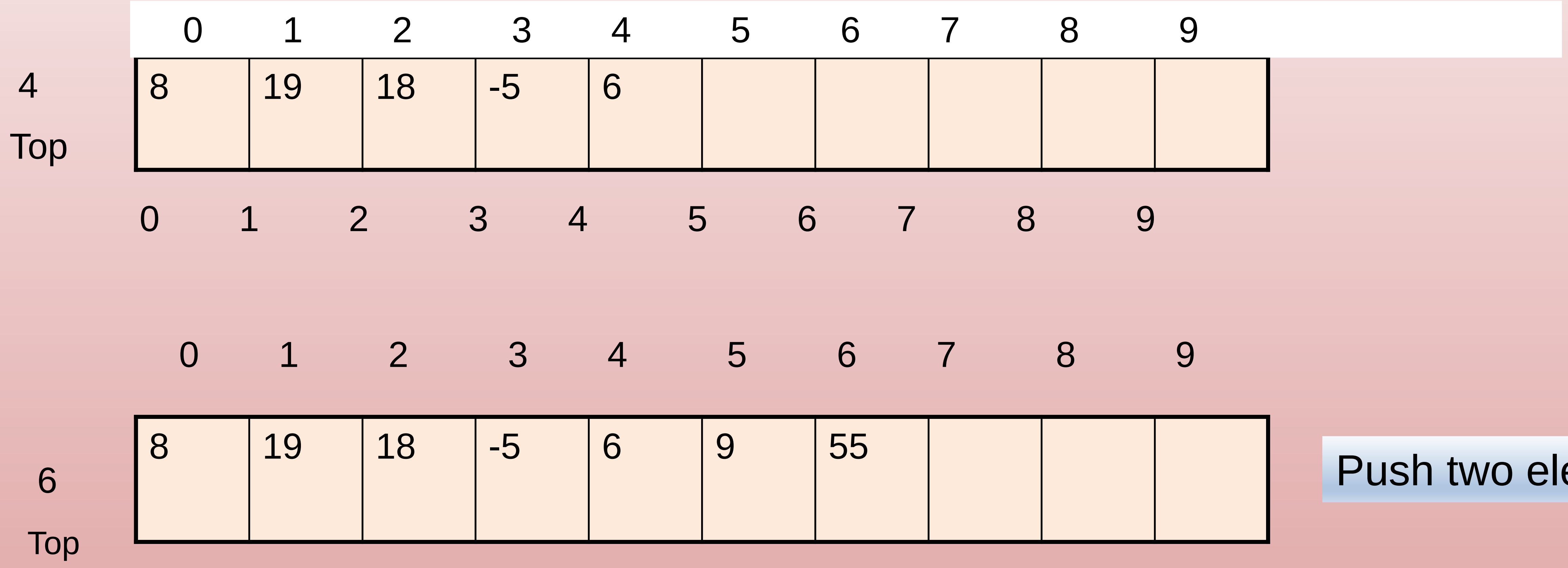
ALGORITHM:

PUSH (STACK, TOP, SIZE, ITEM)

STACK is the array with N elements. TOP is the pointer to the top of the element of the array. ITEM to be inserted.

```
Step 1: if TOP = N then                    [Check Overflow]
              PRINT " STACK is Full or Overflow"
              Exit
        [End if]
Step 2: TOP = TOP + 1                      [Increment the TOP]
Step 3: STACK[TOP] = ITEM                  [Insert the ITEM]
Step 4: Return
```

Representation of Stack in Memory



4

Top

0	1	2	3	4	5	6	7	8	9
8	19	18	-5	6					
0	1	2	3	4	5	6	7	8	9

6

Top

0	1	2	3	4	5	6	7	8	9
8	19	18	-5	6	9	55			

Push two elements => 9, 55

Figure : Push operation in Stack with moving Top pointer

POP Operation

- The process of deleting one element or item from the stack is represented by an operation called as the POP operation.
- When elements are removed continuously from a stack, it shrinks at same end i.e., top

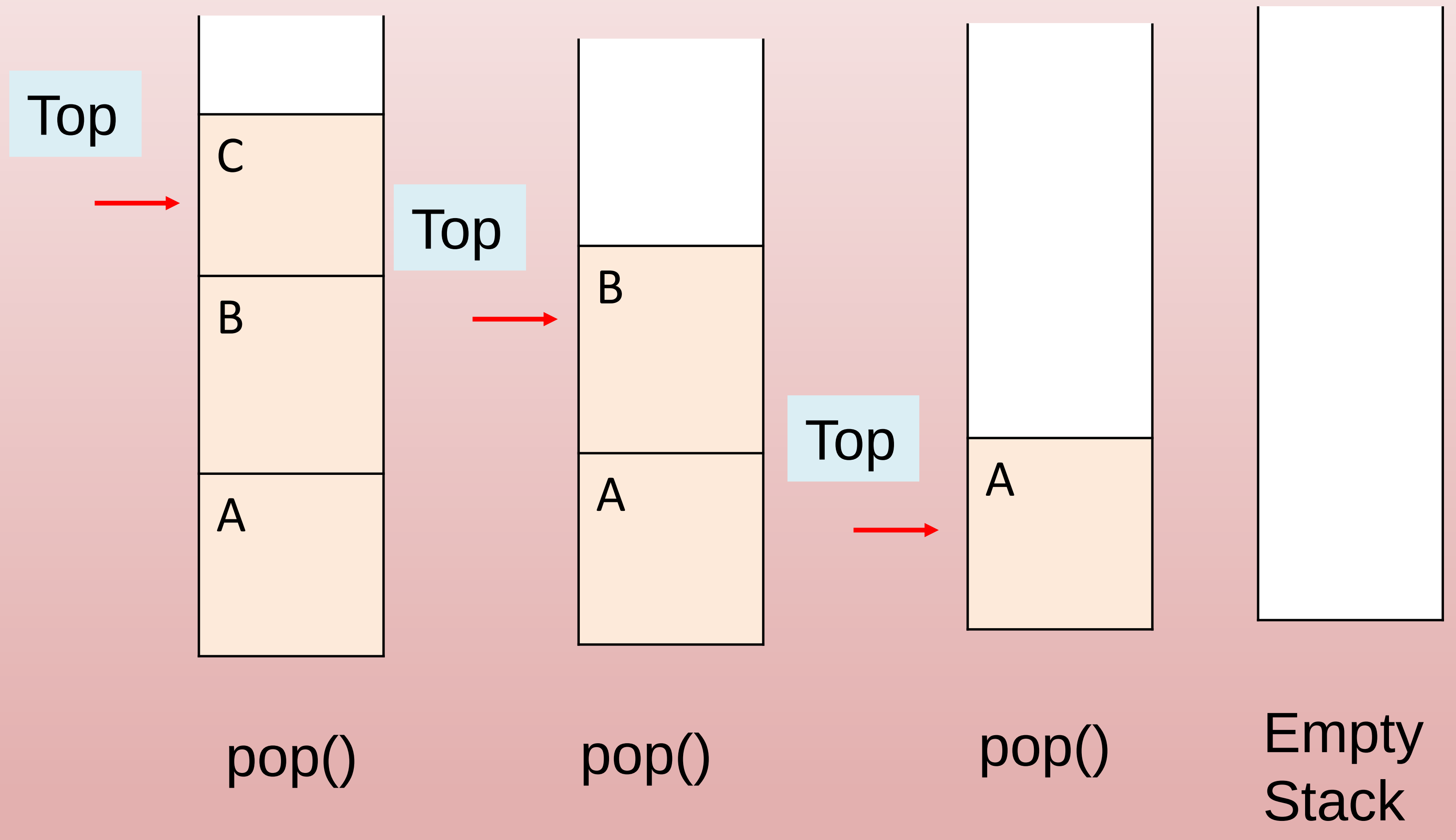


Figure : Pop of Stack

POP Operation Algorithm

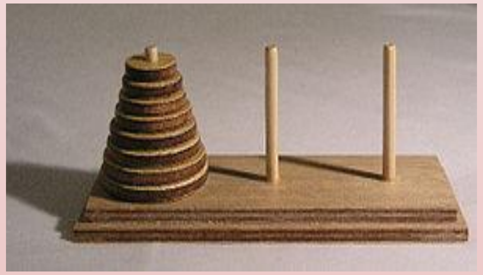
ALGORITHM: POP (STACK, TOP, ITEM)

STACK is the array with N elements. TOP is the pointer to the top of the element of the array. ITEM to be inserted.

```
Step 1: if TOP = 0 then [Check Underflow]
            PRINT “ STACK is Empty or Underflow”
Exit  [End if]
Step 2: ITEM = STACK[TOP] [copy the TOP Element]
Step 3: TOP = TOP – 1         [Decrement the TOP]
Step 4: Return
```

Application of Stacks

- “Undo” mechanism in text editor.
- Conversion of decimal number to binary.
- To solve tower of Hanoi.
- Conversion of infix expression into prefix and postfix.
- Quick sort



What is a Queue?

A queue is a data structure used to model a First-In-First-Out **(FIFO)** strategy. Conceptually, we add to the end of a queue and take away elements from its front.

Figure : Example of Queue

Queue data structure

- Elements added at one end (rear), deleted from other end (front)
- Middle elements inaccessible

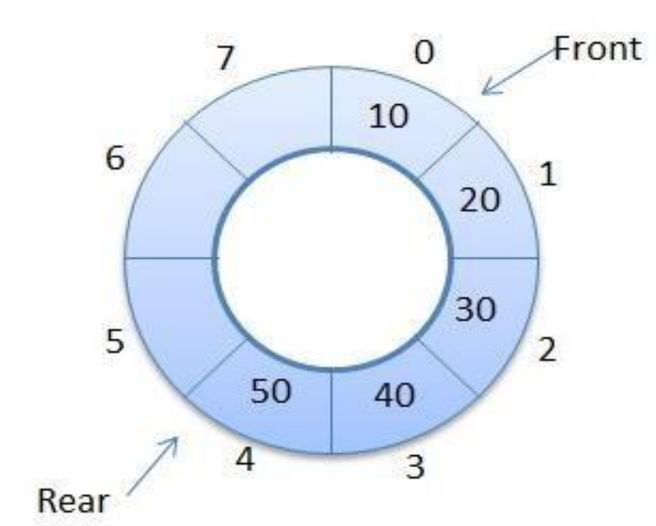
Types of Queue

- A standard queue
- Priority Queue
- Double Ended Queue

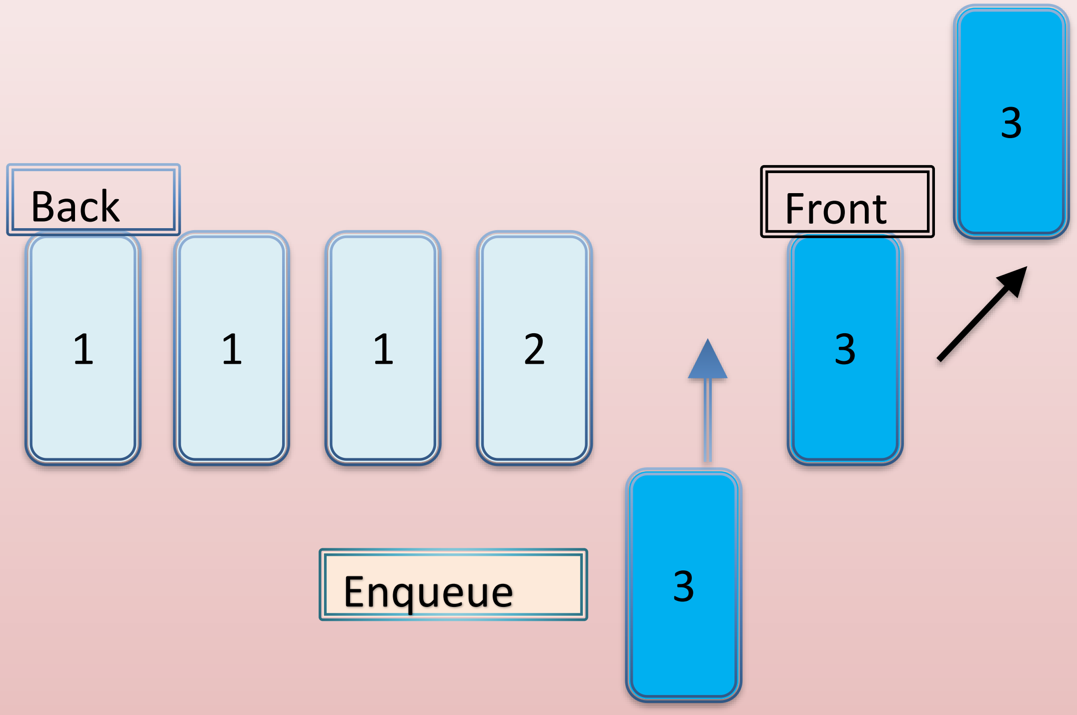
Types of Queues

Queue can be of four types:

1. Circular Queue
2. Priority Queue
3. De-queue (Double Ended Queue)
4. Simple Queue



(a)

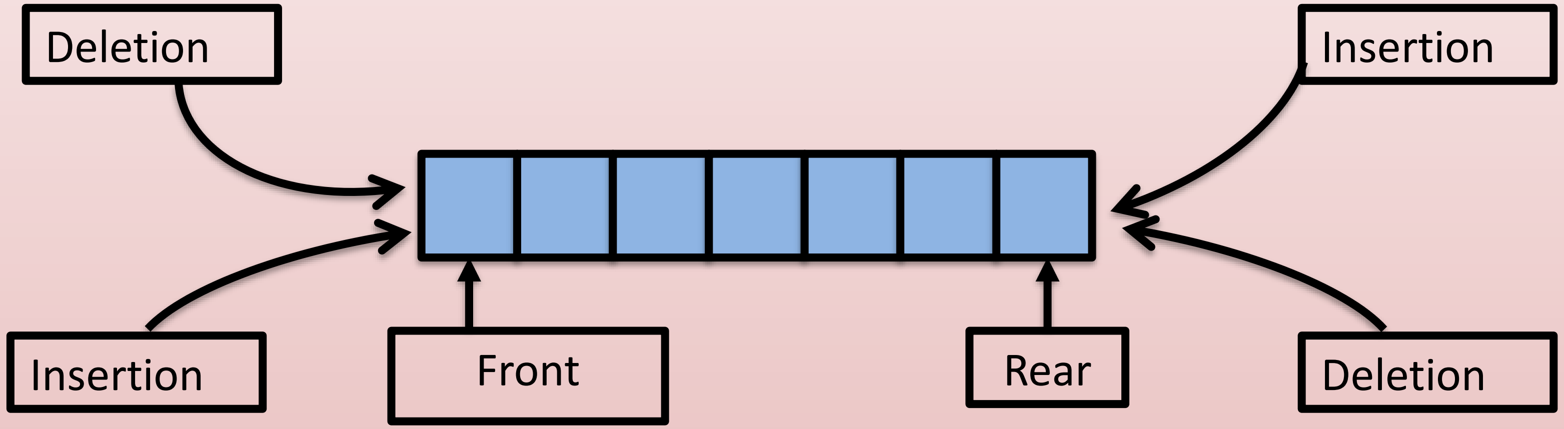


(b)

Figure : Four types of Queue
(a)Circular queue(b) Priority Queue

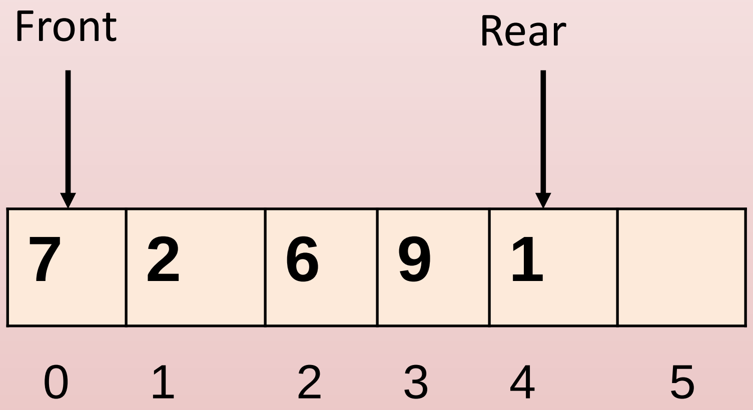
Types of Queues (Cont’d)

Dequeue



(c)

Queue



(d)

Figure : Four types of Queue
(c)De-queue (d) Simple Queue

Implementation of Queues as Arrays

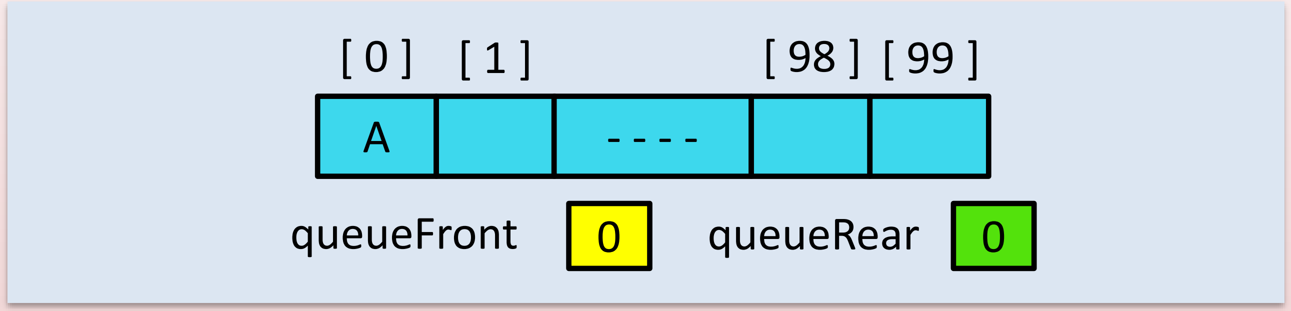


Figure : Queue after the first addQueue operation

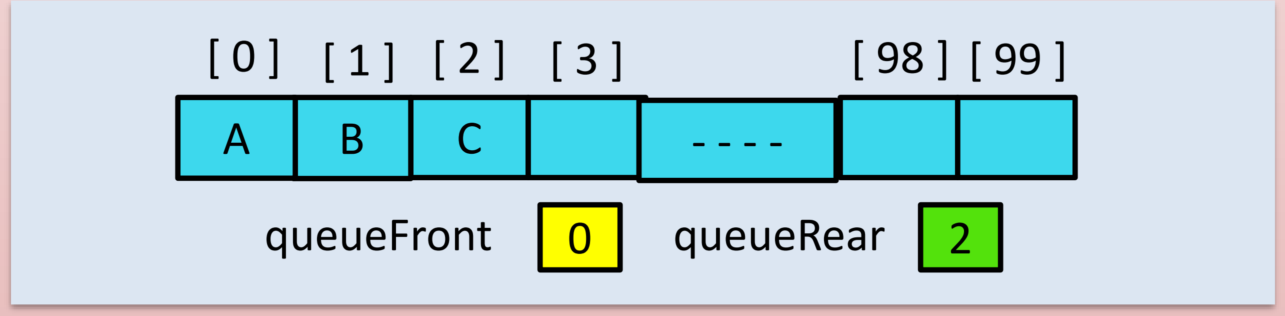


Figure : Queue after two more addQueue operations

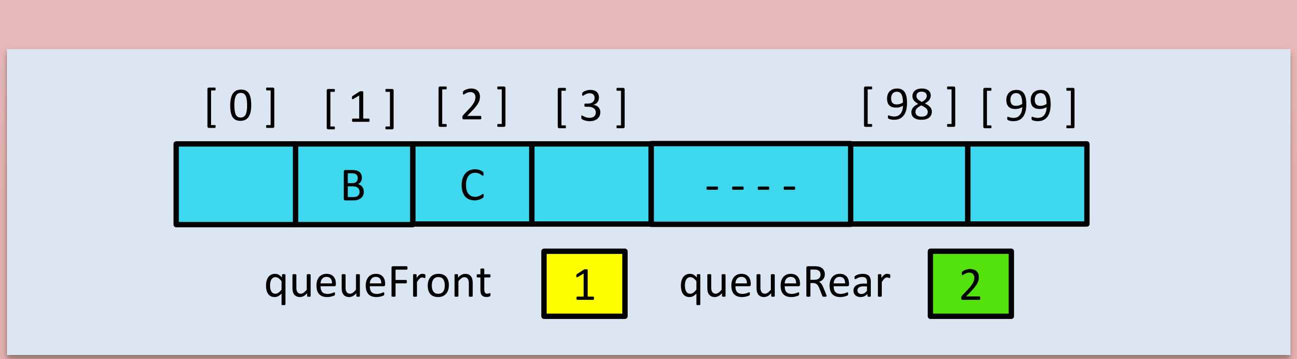


Figure : Queue after the deleteQueue operation

- ➢ Execute operation
  - addQueue(Queue,'A');
- ➢ Execute
  - addQueue(Queue,'B');
  - addQueue(Queue,'C');
- ➢ Execute
  - deleteQueue();

Application of Queues : Simulation

- **Simulation**
  - Technique in which one system models the behavior of another system
- Computer simulation
  - Represents objects being studied as data
  - Actions implemented with algorithms
    - Programming language implements algorithms with functions
    - Functions implement object actions

After learning this lecture, the student will be able to:

- Get the knowledge of memory management of stacks
- Understand the basic operations of stack and queue
- Recognize the allocation of queue as array