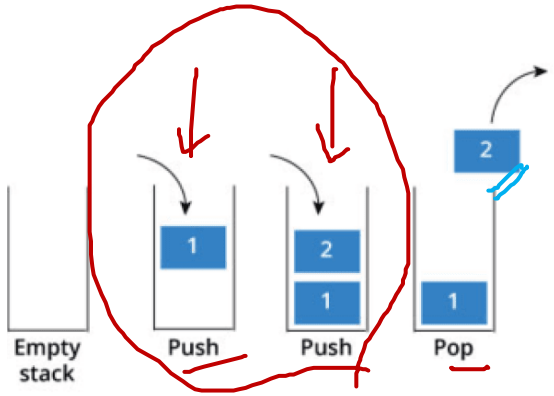
Attendance record

forms.gle/Jv2LFP647Mhawugw7

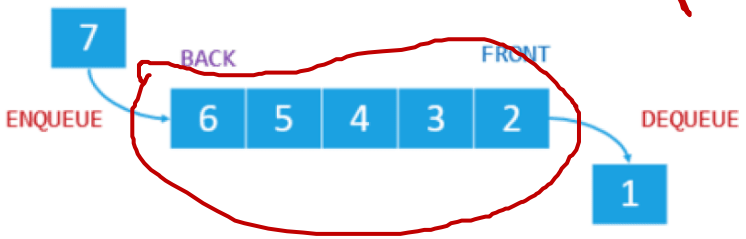
Start: 9:10am

DATA STRUCTURE & PROGRAMMING II

Chapter 10: Queue and Stack



Stack



Queue

Queue

Outline

❑ A Brief of Outline

- What is Queue?
- What are Queue operations?
- How to implement Queue in C++
- Examples

What is Queue?

❑ Definition

- A queue is a data structure that stores data in such a way that the element stored first will be retrieved first
- This method is also called FIFO (First In First Out)

Real life examples:

- A queue of vehicles waiting at the petro pump
- People waiting at the bus store for the bus
- The first person to enter the queue is the first one to leave the queue
- Last person to join the queue is the last person to leave the queue

Applications of Queue

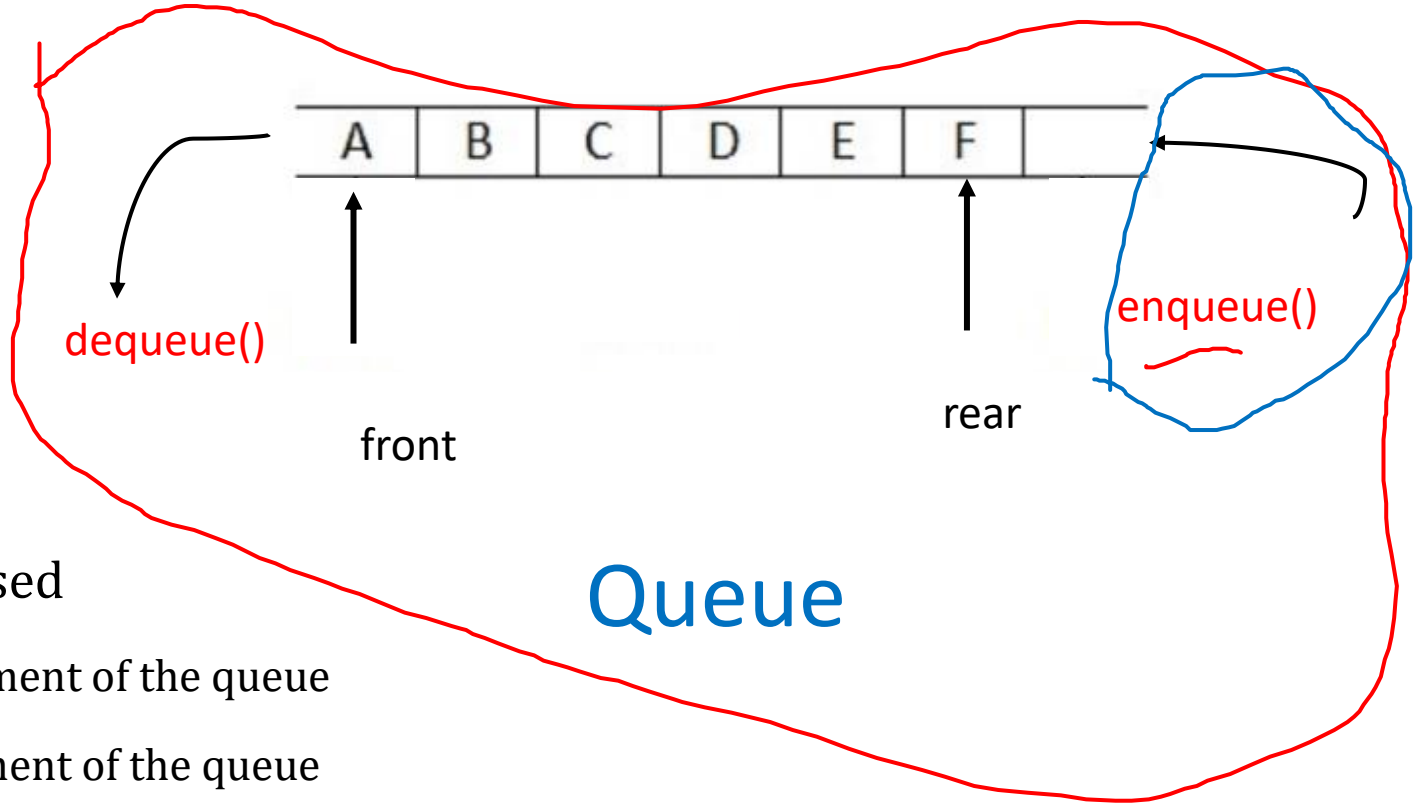
❑ Definition

- Queue finds their use in
  - CPU scheduling,
  - Message queuing,
  - Computer networks
  - etc.
- In time sharing system, queue helps in scheduling of jobs

Queue Operations

❑ Operation

- A queue is controlled by two main operations which implement the FIFO method
  - ***Insertion***
    - Add element to the queue.
    - This method is called ***enqueue***
  - ***Deletion***
    - Remove element from the queue.
    - This method is called ***dequeue***
- Two variables, FRONT and REAR are used
  - FRONT : used for keep track the first element of the queue
  - REAR : used for keep track the last element of the queue



Queue Operations

❑ More operations

- **enqueue**: Add element to end of queue
- **dequeue**: Remove element from front of queue
- **isEmpty**: Check if queue is empty
- **isFull**: Check if queue is full
- **peek**: Get the value of the front of queue without removing it

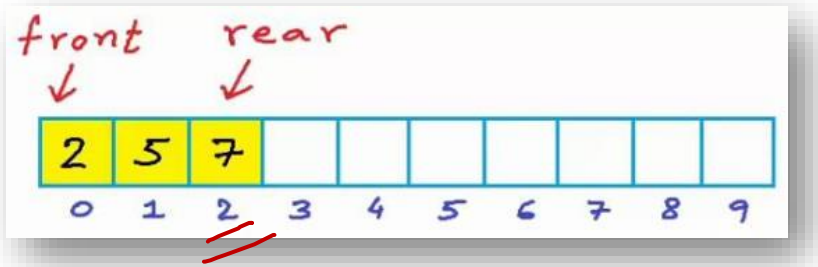
Queue Implementation

❑ Definition

- Queue can be implemented in two ways

1. As an Array

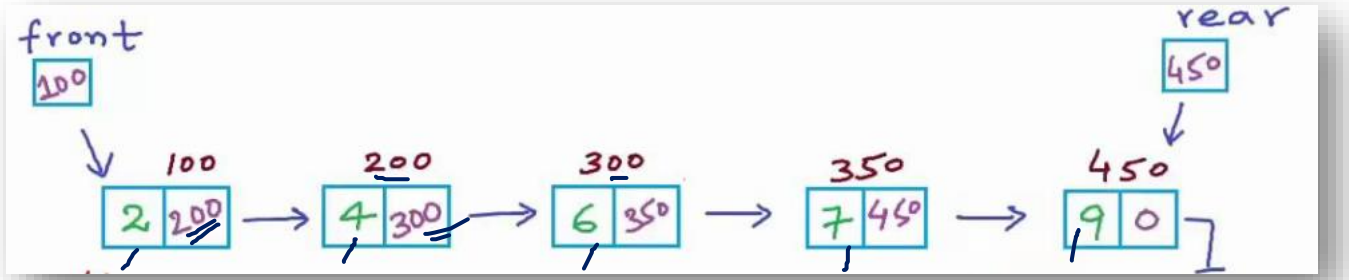
Queue as Array



front variable is used to store the index of the first element
rear variable is used to store the index of the last element

2. As a Linked List

Queue as Linked List



front variable is head of the list
rear variable is tail of the list

Disadvantage of Queue as Array

❑ Definition

- Implementing queue as an array has one major drawback
  - Since arrays are fixed in size, elements can not be inserted beyond the max size of the array

For example:

➢ This queue is considered as full although there are two empty spaces in the beginning of the queue

Implementing Queue as Linked List

Queue Implementation

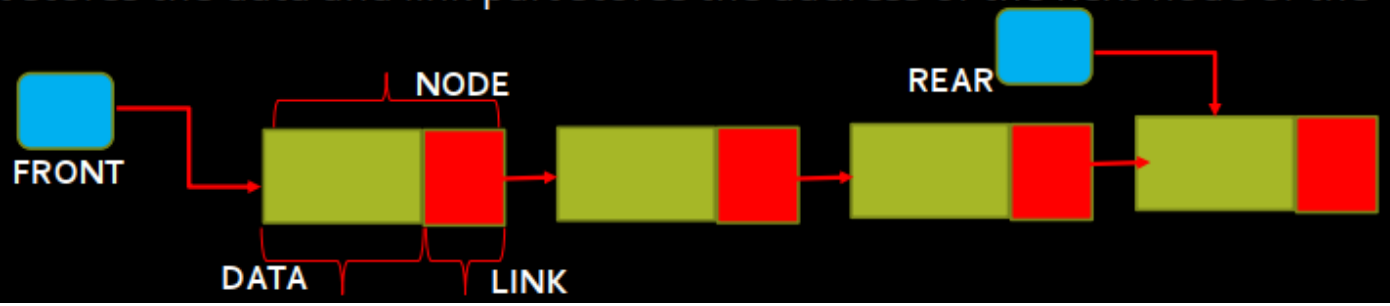
- Queue as a Linked List

QUEUE AS A LINKED LIST

Linked list implementation of queue uses:

- A linked list to store data
- A pointer FRONT pointing to the beginning of the queue and a pointer REAR pointing to the end of the queue.

NOTE: Each node of a queue as a linked list has two parts : data part and link part and is created with the help of self referential structure.
The data part stores the data and link part stores the address of the next node of the linked list.

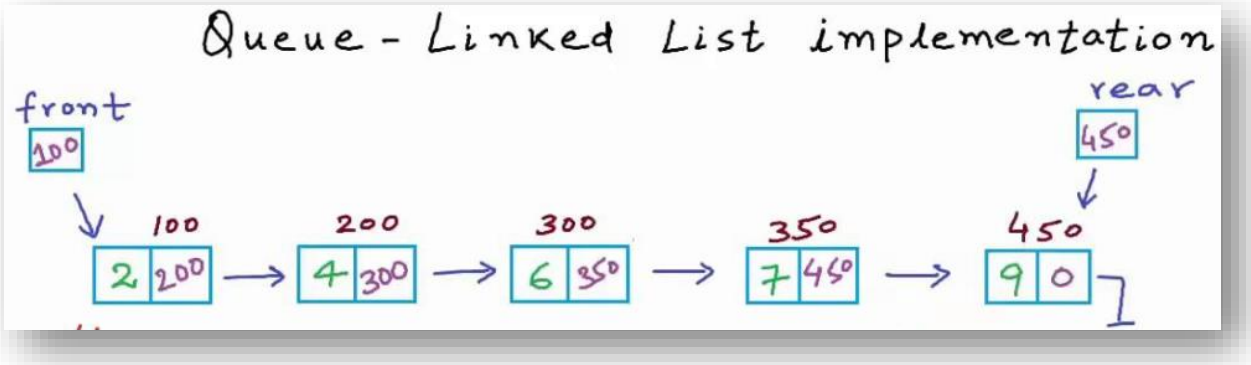


Queue Implementation: Examples

❑ Queue as a Linked List

How to implement this queue?

Demo coding in class



Queue Implementation

❑ Queue as a Linked List

- Implementing queue as a linked list is just like implementing a linked list with some choices

  Choice 1

  - Element is added to the end of the list (*enqueue* operation)
  - Element can be only removed from the beginning of the list (*dequeue* operation)

  Choice 2

  - Element is added to the beginning of the list (*enqueue* operation)
  - Element can be only removed from the end of the list (*dequeue* operation)

Remark: Choice 1 is recommended.

Q and A

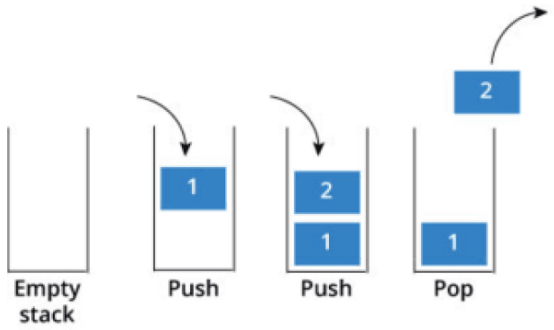
Practice

❑ Exercises

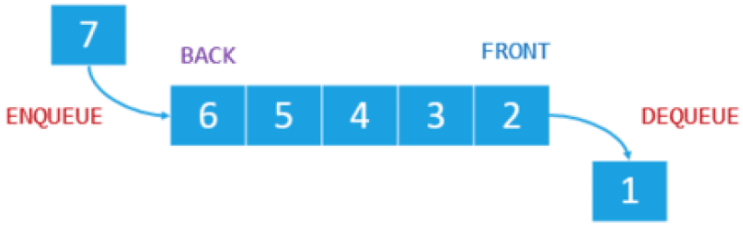
1. Create a queue that stores each letter for an English word input by a user. Then add each letter of this word to this queue.

- ***Ask another user to input a word then test whether a word stored in this queue is the same.***

Stack



Stack

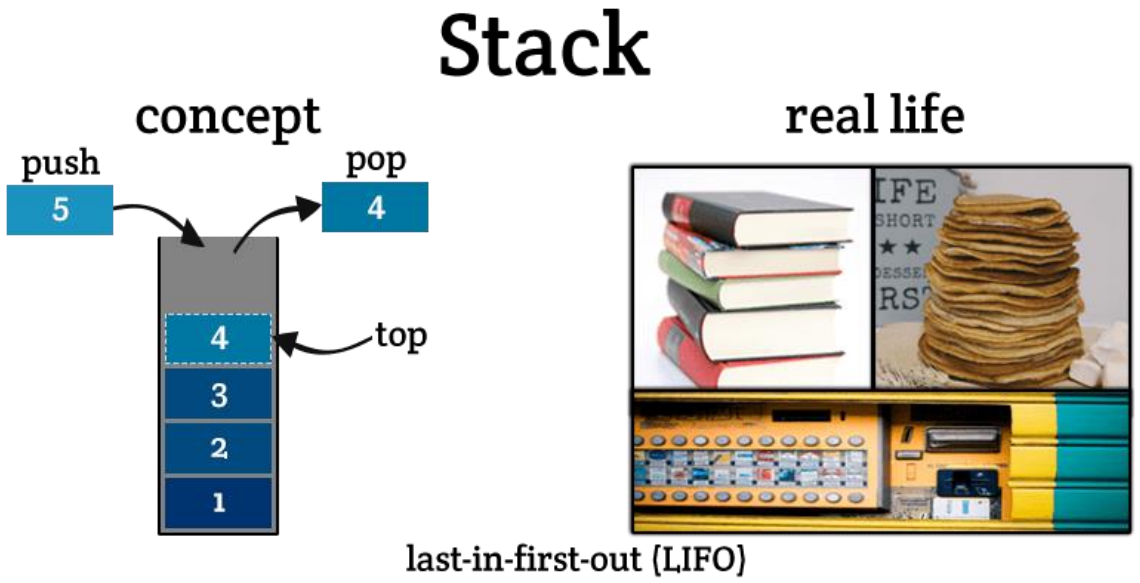


Queue

Outline

❑ A Brief of Outline

- What is Stack?
- What are Stack operations?
- How to implement Stack in C++
- Examples



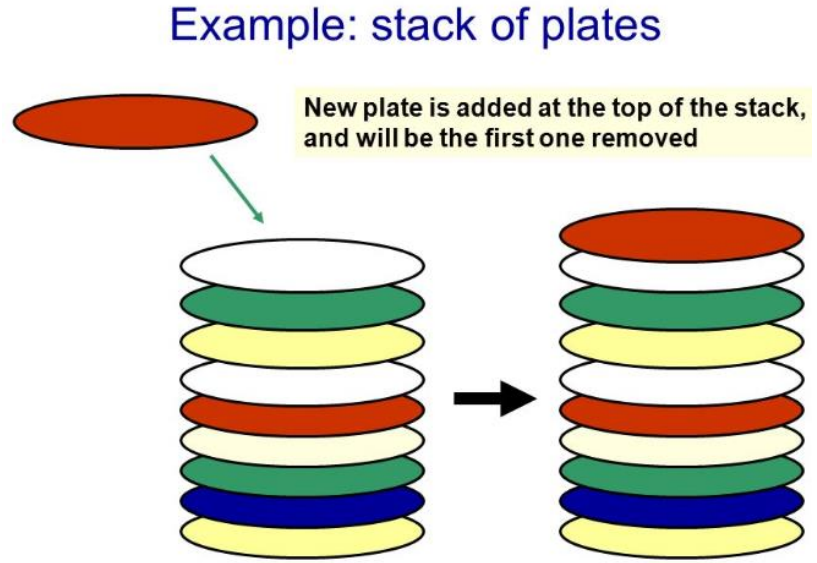
What is Stack?

❑ Definition

- A stack is a data structure that stores data in such a way that the element stored last will be retrieved first
- This method is also called LIFO (Last In First Out)

Examples:

- ➢ A stack of copies
  - The first copy put in the stack is the last one to be removed
  - Similarly, the last copy put in stack is the first one to be removed
- ➢ Stack of plates
- ➢ Stack of chairs

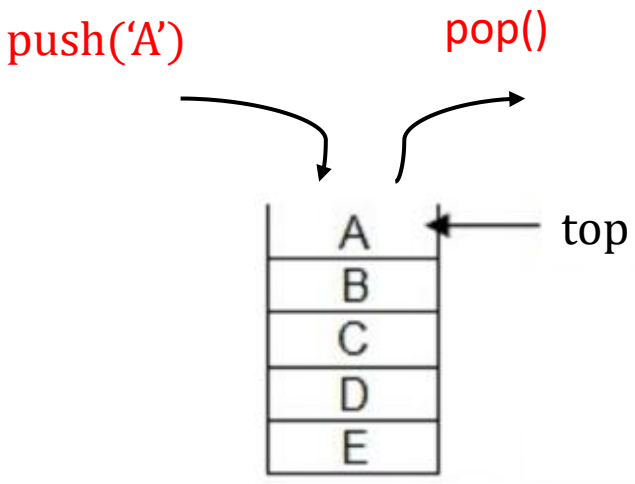


Applications of Stack

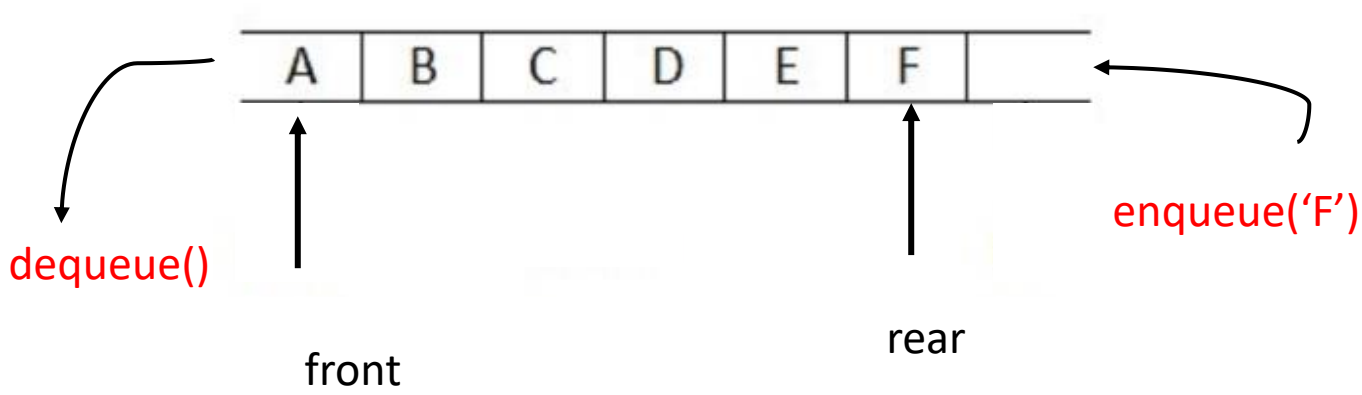
- Undo operation (browser, Ms. Word, …)
- Remembering completed task
- Design compilers and interpreters
- etc.

Queue Vs. Stack

- Differences



Stack

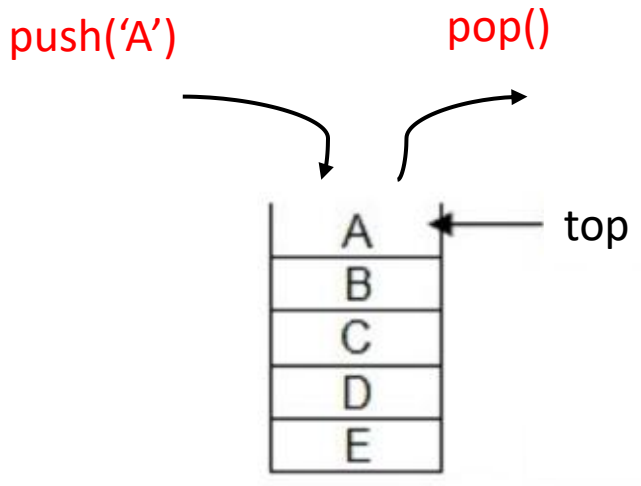


Queue

Stack Operations

❑ Operation

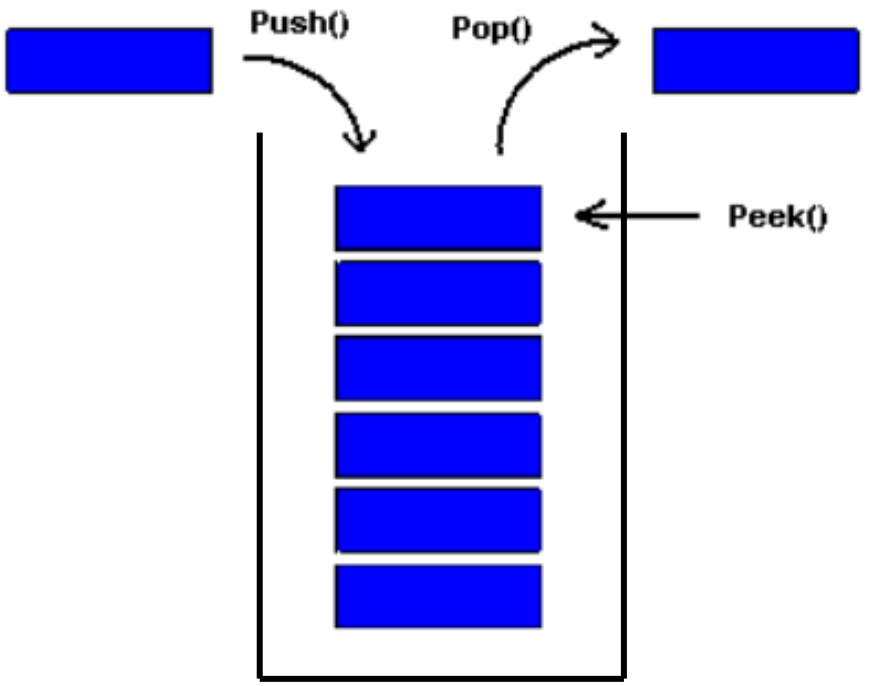
- A stack is controlled by two main operations which implement the LIFO method
  - ***Insertion***
    - Add element to the stack (add to the top)
    - This method is called ***push***
  - ***Deletion***
    - Remove element from the stack (remove from the top)
    - This method is called ***pop***



- The variable TOP is used to keep track of the top element in the stack

Stack Operations

❑ More operations



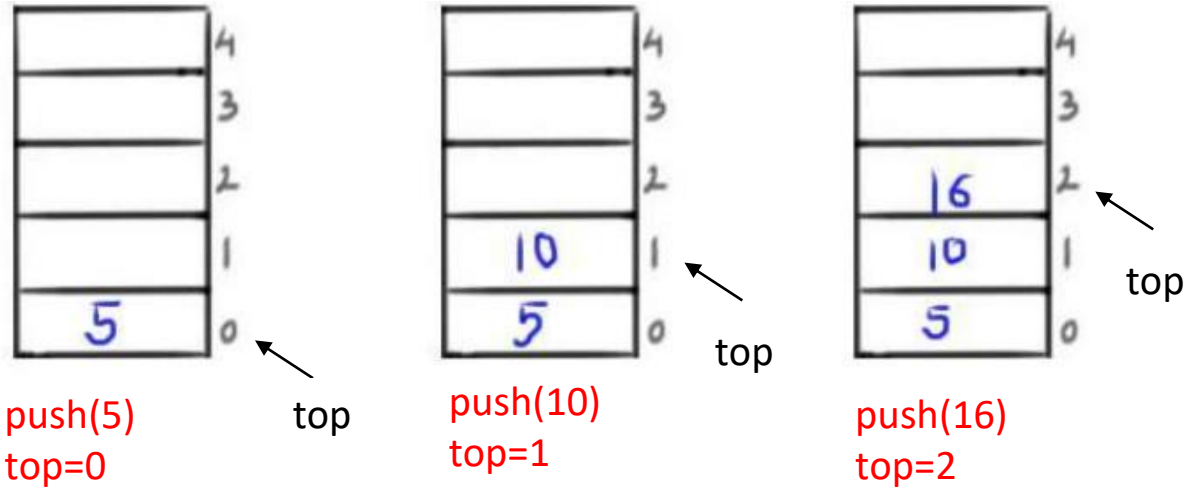
- **`push()`:** Add element to top of stack
- **`pop()`:** Remove element from top of stack
- **`isEmpty()`:** Check if stack is empty
- **`isFull()`:** Check if stack is full
- **`peek()`:** Get the value at the top of stack without removing it

Stack Implementation

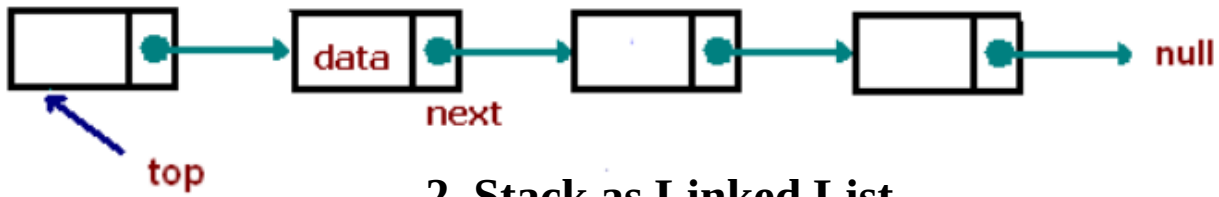
❑ Implementation

➢ Stack can be implemented in two way

1. As an Array
   - An array to store data
   - An integer type variable called TOP which stores the index of the top element of the stack
2. As a Linked List
   - A linked list to store data
   - A pointer variable called TOP which points to the top element of the list



1. Stack as Array



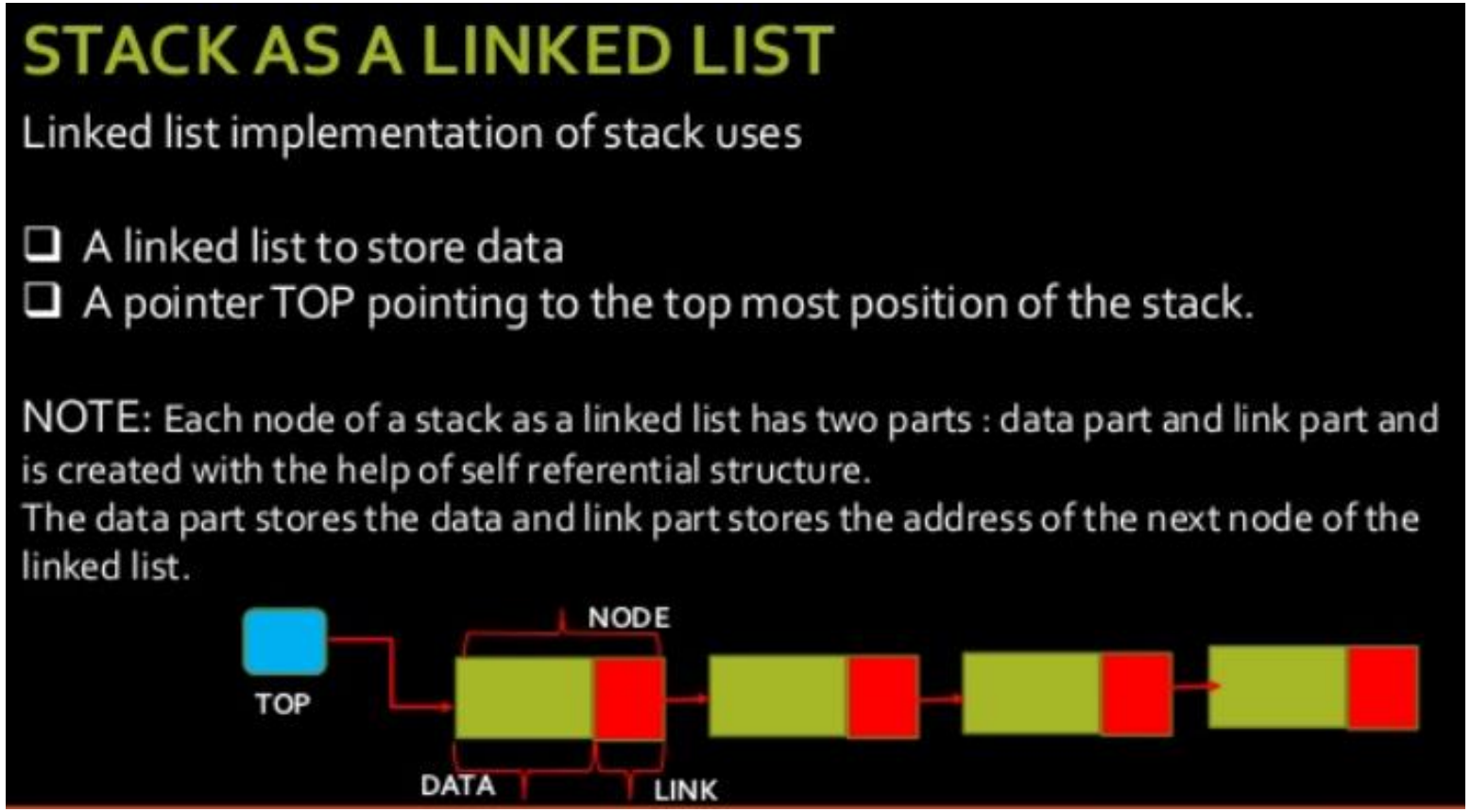
2. Stack as Linked List

Stack as Linked List

- Dynamic
  - It can grow or shrink at runtime

Stack Implementation

- Stack as a Linked List



Stack Implementation

❑ Stack as a Linked List

- Implementing stack as a linked list is just like implementing a linked list with some choices

  Choice 1

  - Element is added to the first of the list **(push operation)**
  - Element can be only removed from the beginning of the list **(pop operation)**

  Choice 2

  - Element is added to the end of the list **(push operation)**
  - Element can be only removed from end of the list **(pop operation)**

Remark: Choice 1 is recommended.

Stack as Array Vs. Stack as Linked list

❑ When to use?

➢ When we use array to implement stack?

- Must know exact size of the stack and it is small

➢ When we use Linked list to implement stack?

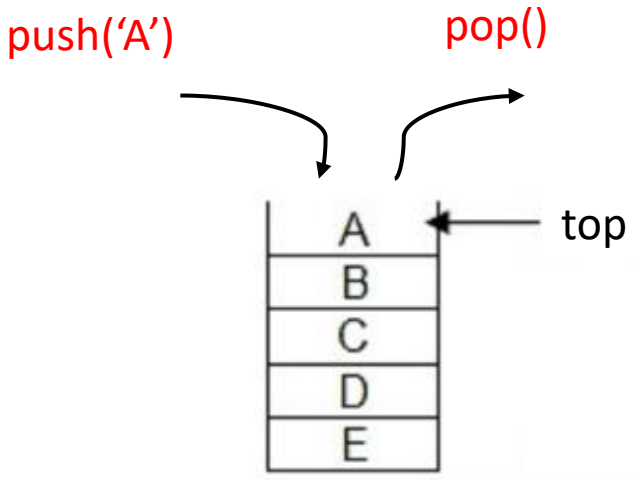
- Do not know the size of the stack.
- If the stack can be bigger and bigger, Linked list will be good to implement it.

Stack Implementation: Examples

❑ Stack as a Linked List

How to implement this Stack?

Demo coding in class



Q and A