



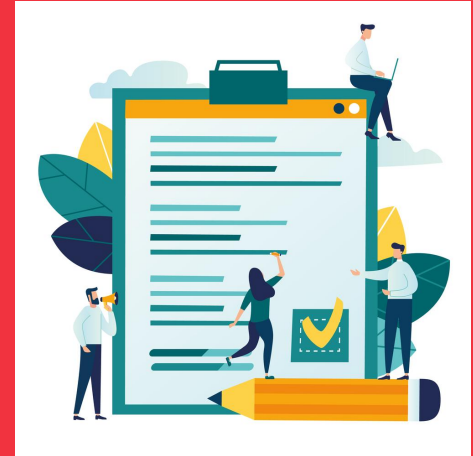
# Data Structures & Algorithms

**Course:** Data Structures and Algorithm

**Lecture On:** Stacks

# In the previous class, we covered....

- Circular Linked List
- Doubly Linked List



# Homework Discussion

1. Reverse a Circular LL

# Today's Agenda

## 1 Stacks



## Stacks

How do we see trays piled up in a cafeteria today? Or a tube of Pringles when we open it up? We see that they are **'stacked'** one over the other. To be able to eat a Pringle in the middle of this 'stack', we have to take out or eat all the Pringles on top of it first.

This idea forms the basic construct of the 'Stack' ADT (Abstract Data Type). As we can understand, a Stack is a linear data structure, as the data (chips) are placed sequentially or in a linear pattern in the data structure (can of Pringles).



## Stacks

As we understand, a Stack is very similar to an Array, but with one restriction. We can only insert data and take data out of the Stack from the **'top'** of the Stack. Formally, these operations are known as **'push'** (inserting data) and **'pop'** (taking data out). As we can understand, at any given time, we can access only one data element in the data structure.

This unique feature of stack makes it a **'LIFO'** data structure, which stands for **'Last In, First Out'**. As the name suggests, the data element that is 'last in' will be the one that will be the 'first out' of the data structure.

## Stacks

Although **'Push'** and **'Pop'** are the most important and common Stack operations, the following are a few other operations that we can perform on Stacks:

- **peek** – To 'peek' at the top element in the Stack
- **isEmpty** – To check if the Stack is empty or not
- **isFull** – To check if the Stack is full or not



# Poll 1 (15 Sec.)

Which of the following is a linear data structure?

1. Stack
2. Array
3. Linked List
4. All of the above

# Poll 1 (Answer)

Which of the following is a linear data structure?

1. Stack
2. Array
3. Linked List
4. **All of the above**

## Poll 2 (15 Sec.)

Which of the following is a FILO or LIFO data structure?

1. Linked List
2. Stack
3. Array
4. All of the above

# Poll 2 (Answer)

Which of the following is a FILO or LIFO data structure?

1. Linked List
2. **Stack**
3. Array
4. All of the above

## Poll 3 (15 Sec)

Which stack operation can be used to fetch the top most element in the stack?

1. push
2. peek
3. isEmpty
4. All of the above

## Poll 3 (Answer)

Which stack operation can be used to fetch the top most element in the stack?

1. push
2. **peek**
3. isEmpty
4. All of the above

## Poll 4 (15 Sec)

Which stack operation can be used to check if the stack is full?

1. push
2. isFull
3. isEmpty
4. peek



# Poll 4 (Answer)

Which stack operation can be used to check if the stack is full?

1. push
2. **isFull**
3. isEmpty
4. peek



## Stack Implementation

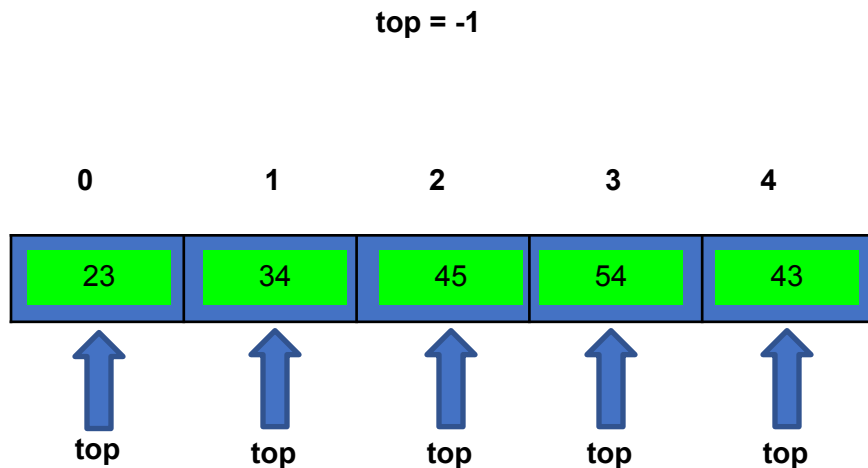
We can use Arrays or Linked Lists to implement the behaviour of a Stack. **Let's explore how we can do this.**

To implement a Stack using an Array, we will need to declare an array in advance (with a fixed size). We would also need to maintain a variable to keep track of the 'top' of the Stack.

As the data gets pushed in or popped out of the array, the 'top' tracker will 'move' along the indices to point to the new 'top' of the array. At the start, we initialise the 'top' with a value of -1 representing absence of any value in a stack.

## Stack of five elements created:

1. Pop (underflow)
2. Push 23
3. Push 34
4. Peek (returns 34)
5. Push 45
6. Push 56
7. Pop (returns 56)
8. Push 54
9. Push 43
10. Push 32 ( $\text{top}+1 > n-1$ , overflow)
11. Pop



## Poll 5 (15 Sec)

Given the stack from the top: 8 9 10 11 12 13 14 15 16 (16 is at the bottom of the stack)

What would be the output after the following operations:

Push(20)

Pop

Pop

Peek

1. 8

2. 9

3. 10

4. 20

## Poll 5 (Answer)

Given the stack from the top: 8 9 10 11 12 13 14 15 16 (16 is at the bottom of the stack)

What would be the output after the following operations:

Push(20)

Pop

Pop

Peek

1. 8

2. 9

3. 10

4. 20

## Hands-on Coding

- [Write a program to implement Stack using Array](#)
- [Write a program to implement Stack using Singly Linked List.](#)

## Poll 6 (15 Sec.)

What is the time complexity of Stack operations *push()* and *pop()* when implemented using singly linked list?

1.  $O(n)$
2.  $O(1)$
3.  $O(n^2)$
4. None of these

# Poll 6 (Answer)

What is the time complexity of Stack operations *push()* and *pop()* when implemented using singly linked list?

1.  $O(n)$
2.  **$O(1)$**
3.  $O(n^2)$
4. None of these

## Poll 7 (15 Sec.)

Stacks can be used for which of the following operations?

1. Support 'undo' operations in several text editors
2. Bracket balancing
3. Stock market algorithms
4. All of the above



## Poll 7 (Answer)

Stacks can be used for which of the following operations?

1. Support 'undo' operations in several text editors
2. Bracket balancing
3. Stock market algorithms
4. **All of the above**

## Hands-on Coding

- [Sort a stack](#)
- [Remove unbalanced parentheses in a given expression.](#)
- [Implement two stacks using a single array](#)

# Homework

1. Write a program that would reverse a stack

# Tasks to complete after the session

Homework
MCQs
Coding Questions



Thank You!