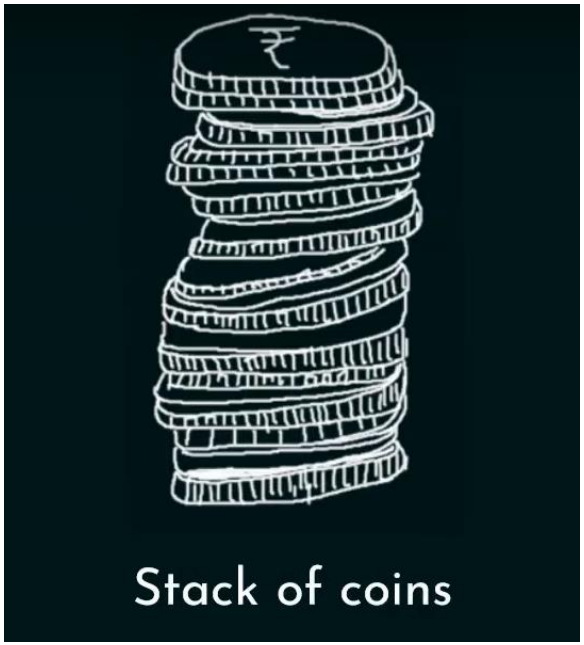
Data Structures and Algorithms

Lecture 15: Stacks

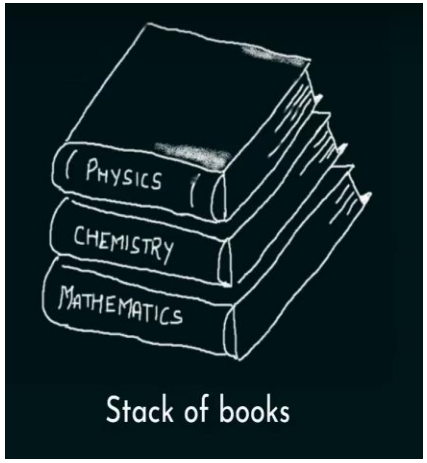
Stack - Introduction

Stack is a linear data structure in which insertion & deletion is allowed at one end called top.



Stack - Introduction

The order may be LIFO(Last In First Out) or FILO(First In Last Out).

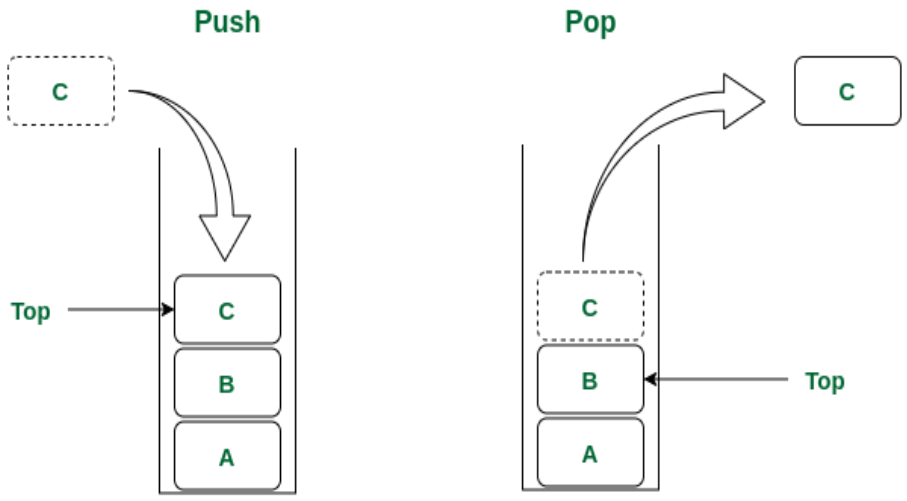


Stack of books

LIFO implies that the element that is inserted last, comes out first and FILO implies that the element that is inserted first, comes out last.

Stack Primary Operations:

- push(data) - inserts data onto the Stack
- pop() - deletes the last inserted element from the Stack



Stack Data Structure

Commands:

1. Push - Adds an item to the stack. If the stack is full, then it is said to be an Overflow condition.
2. Pop - Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
3. Top - Returns the top element of the stack.
4. isEmpty - Returns true if the stack is empty, else false.

Stack Secondary Operations:

- top() - returns the last inserted element without removing it
- size() - returns the size or the number of elements in the stack
- isEmpty() - returns TRUE if Stack is empty. Else returns FALSE
- isFull() - returns TRUE is Stack is full. Else returns FALSE

Implementation:

- Stack is a linear data structure that operates on the LIFO principle and can be implemented using an array or a linked list.
- The basic operations that can be performed on a stack include push, pop, and peek, and stacks are commonly used in computer science for a variety of applications, including the evaluation of expressions, function calls, and memory management.
- There are two ways to implement a stack –
  - Using array
  - Using linked list

Stack Applications:

- Infix to Postfix evaluation
- Infix to Postfix conversion
- Nested Brackets
- Balanced Parenthesis
- Undo-Redo
- Memory Management

Stack Applications:

- Infix

  Example 1 : A + B

  Example 2 : A * B + C / D

- Postfix

  Example 1 : AB+

  Example 2 : AB*CD/+

- Prefix

  Example 1 : +AB

  Example 2 : +*AB/CD

Recall:

- Stack
- Push
- POP
- Implementation
- Applications