I am going to talk about stack data structure

Stack is a simple data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the top of the stack and only the top element can be removed stack, just like a real-world stack, piles of books, a deck of cards, or a pile of plates, etc.

Stack is a LIFO(Last in First out) structure. Meaning that the element which was added last will be the first element to be removed.

## Now lets talk about some of the basic Basic Operations on stack

a stack is used for the following two primary operations −

* push() − storing an element on the stack.
* pop() − Removing or accessing an element from the stack.

Before going in details lets see some other operations that are used to check the status of a stack

1st one is

* peek() − get the top data element of the stack, without removing it.
* isFull() − check if stack is full.
* isEmpty() − check if stack is empty.

At all times, we maintain a pointer to the last PUSHed data on the stack. As this pointer always represents the top of the stack. The top pointer provides the top value of the stack without actually removing it.

## Push Operation

The process of putting a new data element onto the stack is known as a Push Operation. Push operation involves a series of steps −

1. Check if the stack is full or not.
2. If the stack is full, then print the error of overflow and exit the program.
3. If the stack is not full, then increment the top and add the element.

## Pop Operation

Accessing the content while removing it from the stack, is known as a Pop Operation. A Pop operation may involve the following steps −

1. Check if the stack is empty or not.
2. If the stack is empty, then print the error of underflow and exit the program.
3. If the stack is not empty, then print the element at the top and decrement the top.

### Analysis of Stack Operations

Below mentioned are the time complexities for various operations that can be performed on the Stack data structure.

* Push Operation : O(1)
* Pop Operation : O(1)
* Top Operation : O(1)
* Search Operation : O(n)