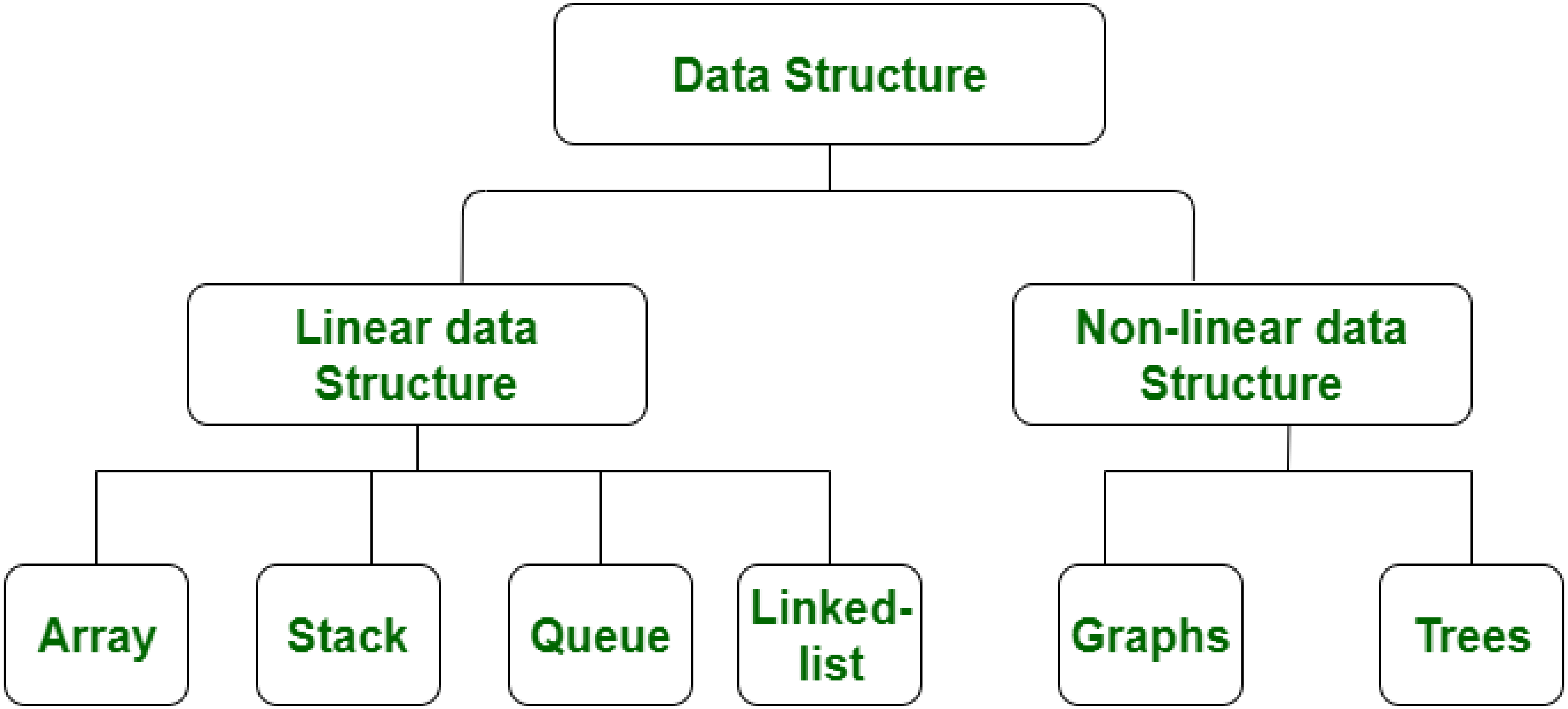
Data Structure

ENCS205

School of Engineering & Technology (SOET)
K.R. MANGALAM University

UNIT-2
Session 22: Stack ADT and Uses

RECAP



https://www.geeksforgeeks.org/data-structure-meaning/

Session 22 Outlook

- Background
- Basic Introduction of stack
- Operations of stack
- Working of Stack
- Uses

Key Learning Outcomes

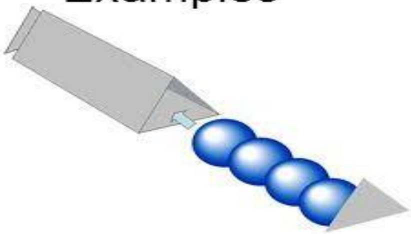
- Students should be able to recall stack terminology and ADT operations.
- Student should be able to Understand stack structure vs. other data structures. Differentiate implementations and expression representations.
- Apply array-based stack operations and conversion algorithms.
- Students should evaluate efficiency of stack operations and algorithms. Evaluate impact of stack structure on complexity.

Data Structure

Stack

Examples

Stack of Books



Box of Tennis Balls
(Closed at 1 end)

Stack of Plates

Stack of Clothes.

Trains in a railway yard.

Goods in a cargo.

Type:

LIFO / FILO

Stack of Rings / Discs

Stack of Chairs

Background

"stack n.

The set of things a person has to do in the future. "I haven't done it yet because every time I pop my stack something new gets pushed." If you are interrupted several times in the middle of a conversation, "My stack overflowed" means "I forget what we were talking about."

-The Hacker's Dictionary

Friedrich L. Bauer
German computer scientist
who proposed "stack method
of expression evaluation"
in 1955.

Structure of Stack?

A structure is like a stack of plates—items can only be added or removed from the top.

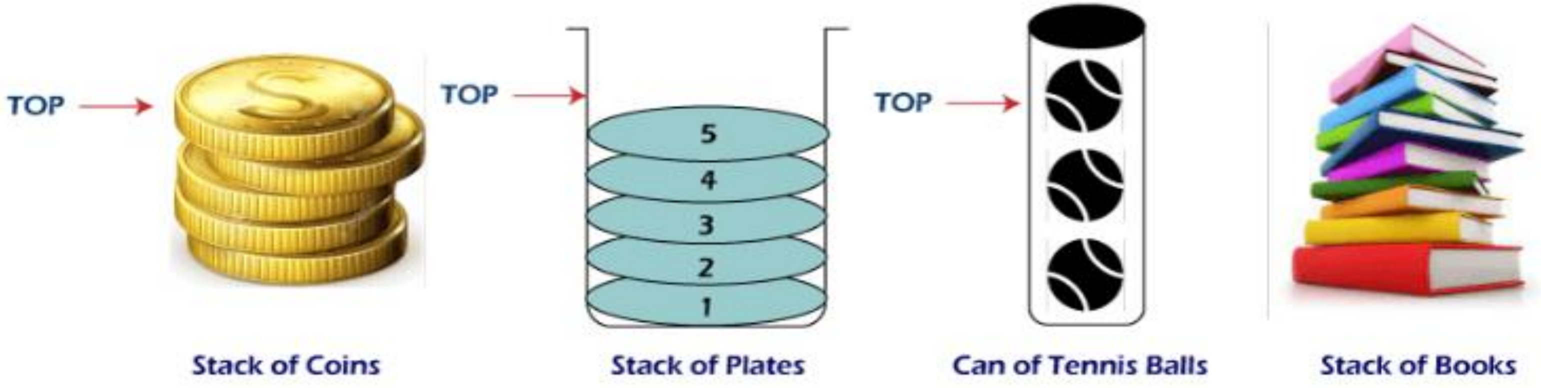
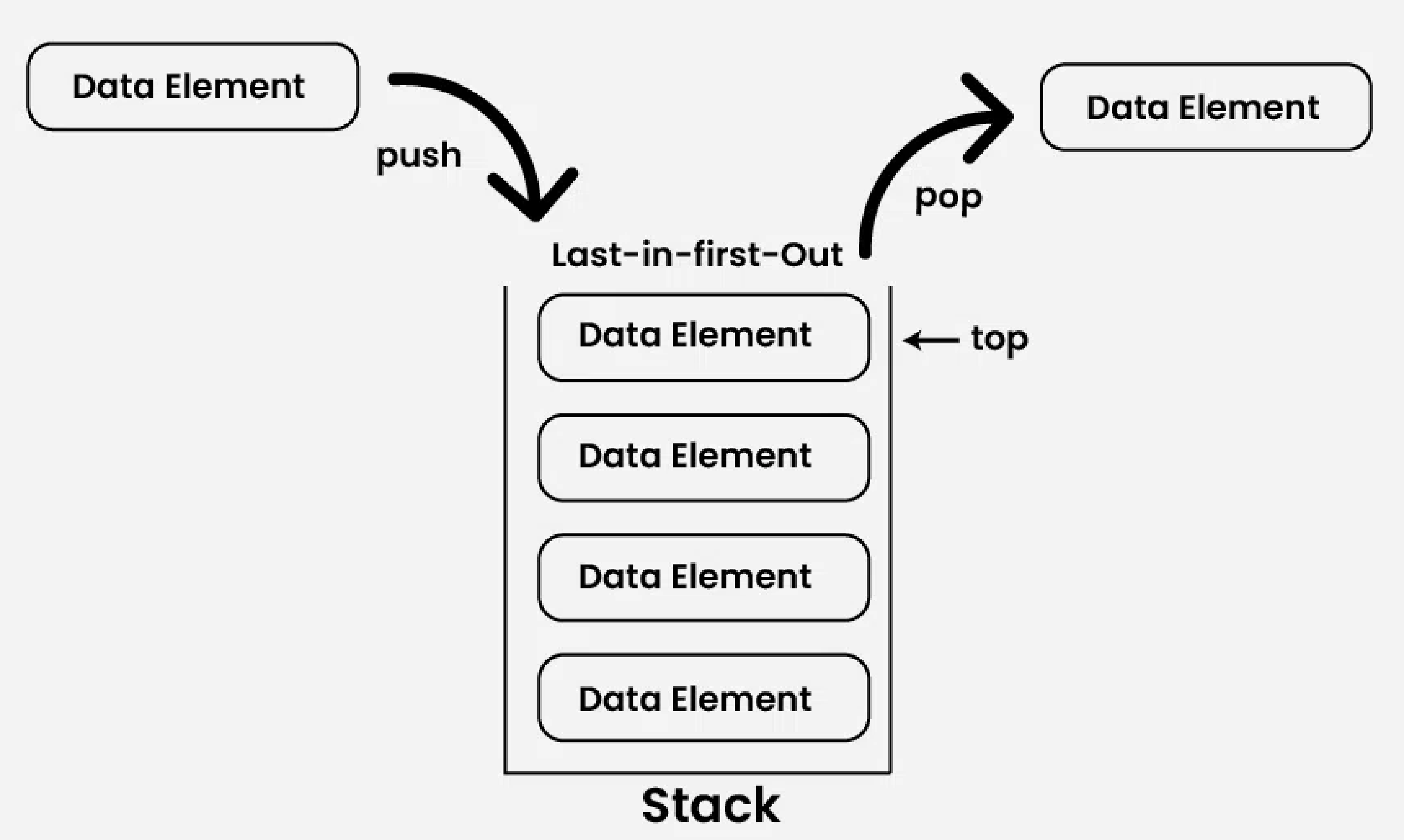


Fig 1 : Stack Data Structure
https://www.javatpoint.com/applications-of-stack-in-data-structure

Stack?

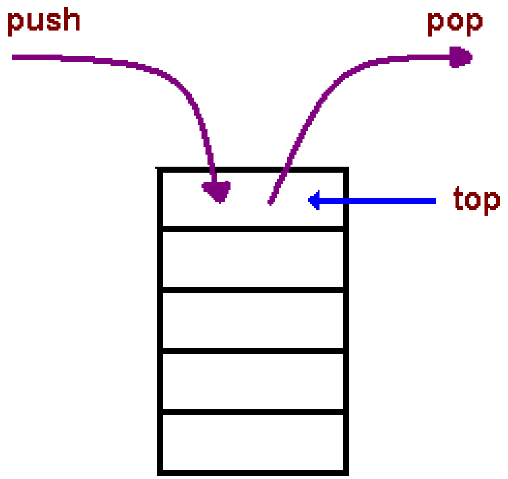


https://www.geeksforgeeks.org/dsa/introduction-to-stack-data-structure-and-algorithm-tutorials/

Stack?

- A Stack is a linear data structure that holds a linear, ordered sequence of elements. It is an abstract data type.
- A stack is basically a container where items are inserted and removed from one end, called the top of the stack..

Some key points related to stack:



Real-world Analogy

Limited Capacity

Order of Operations(LIFO) or (FILO).

Abstract Data Type

https://dev.to/ogwurujohnson/demystifying-the-javascript-call-stack-1ppc

Types of Stack

Fixed Size Stack :

- Fixed size and cannot grow or shrink dynamically.

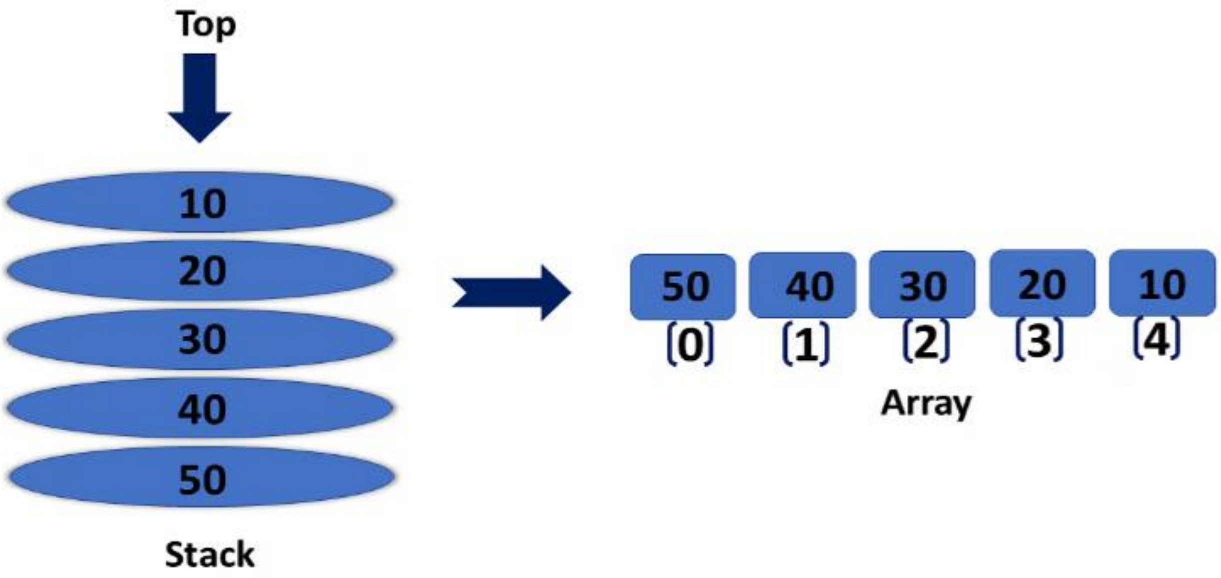
Dynamic Size Stack :

- Can grow or shrink dynamically.
- When the stack is full, it automatically increases its size to accommodate the new element, and when the stack is empty, it decreases its size.
- This type of stack is implemented using a linked list, as it allows for easy resizing of the stack.

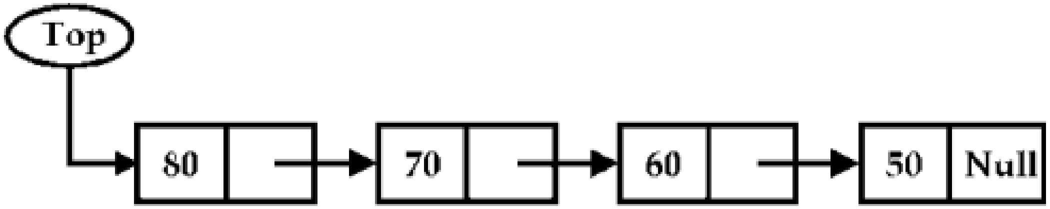
https://www.geeksforgeeks.org/dsa/introduction-to-stack-data-structure-and-algorithm-tutorials/

REPRESENTATION OF A STACK

- Array Representation of Stacks



- Linked List Representation of Stacks



Stack Operations

push() to insert an element into the stack

pop() to remove an element from the stack

peek() or top() returns the topmost element of the Stack without deleting it from the Stack.

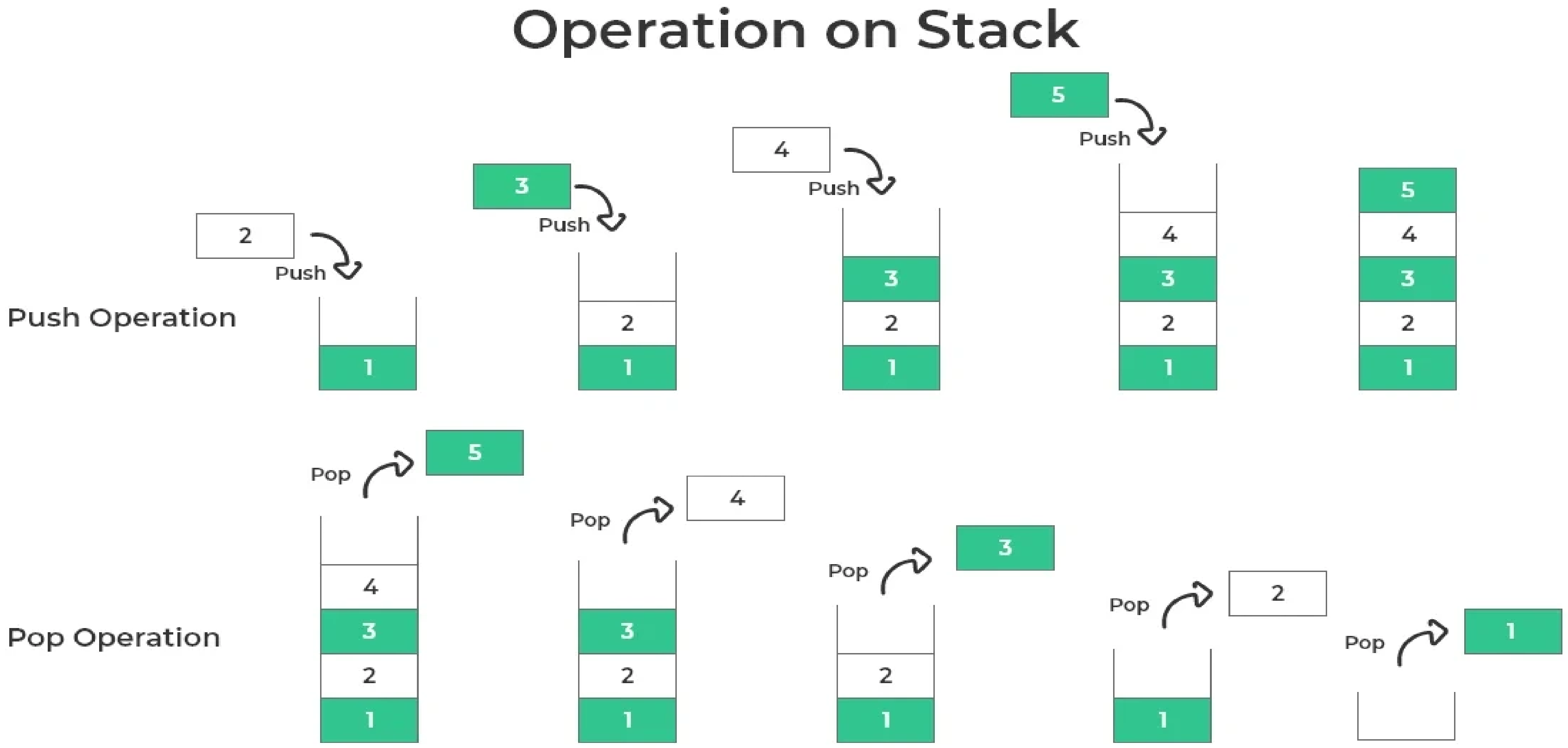
Is Empty() returns true if stack is empty else false.

size() returns the size of stack.

Please Refer :

https://www.youtube.com/watch?v=HRDHgKrYHgU

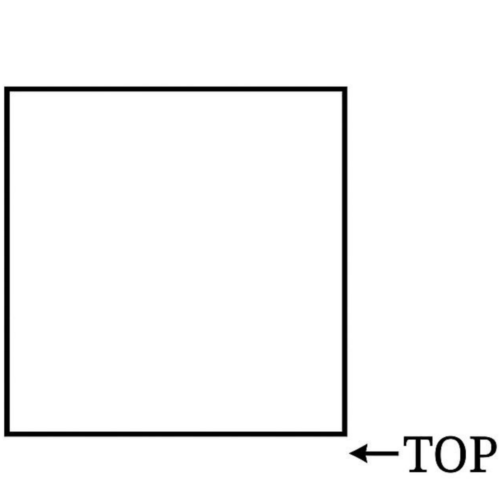
Stack Operations Push & Pop



https://prepinsta.com/data-structures/introduction-to-stacks-in-data-structures/

Conditions in Stack

- Overflow
- Underflow



(a) Underflow (b) Overflow

https://www.geeksforgeeks.org/dsa/stack-data-structure/

PUSH Operation

```
Algorithm:
Step-1: If TOP = Max-1
Print “Overflow”
Goto Step 4
Step-2: Set TOP= TOP + 1
Step-3: Set Stack[TOP]= ELEMENT
Step-4: END
```

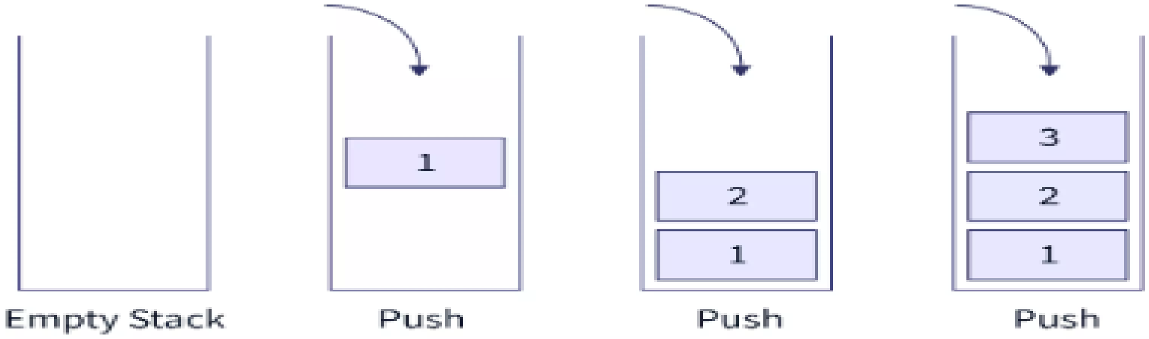


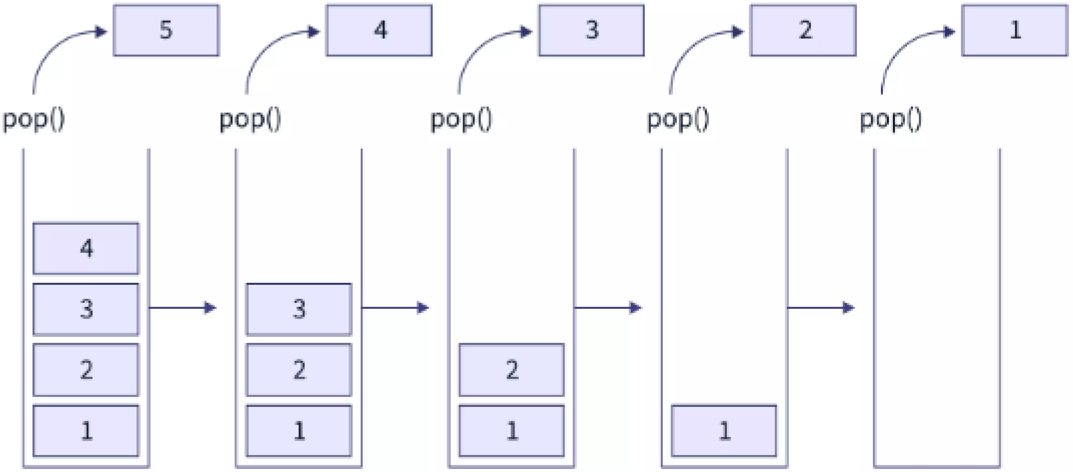
Fig: Push operation

https://www.scaler.com/topics/stack-operations-in-data-structures/

Pop Operations

Algorithm:

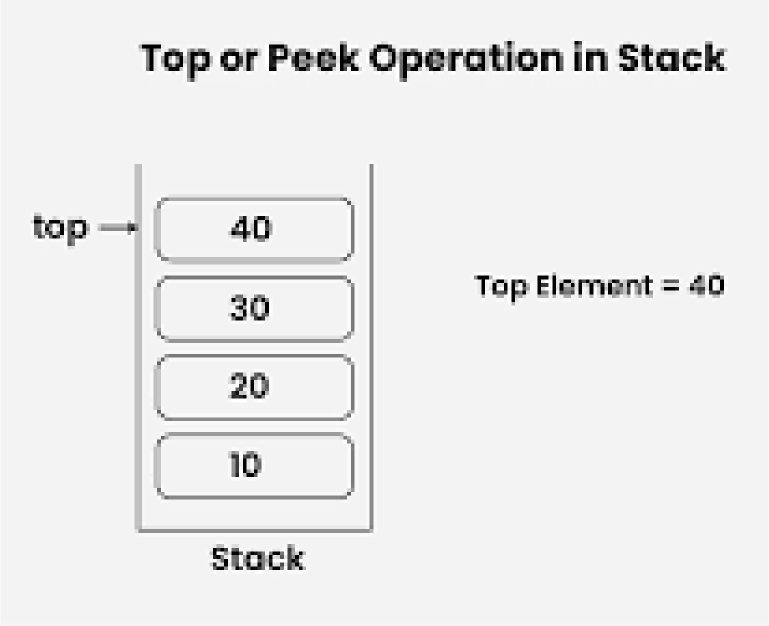
```
Step 1: START
Step 2: if top==-1
then Write “Stack is Underflow”
Step 3: otherwise
        print “deleted element”
         top=top-1;
Step 4: END
```



Top of Peek Operation

Algorithm:

```
Step-1: If TOP = NULL
PRINT “Stack is Empty”
Goto Step 3
Step-2: Return Stack[TOP]
Step-3: END
```

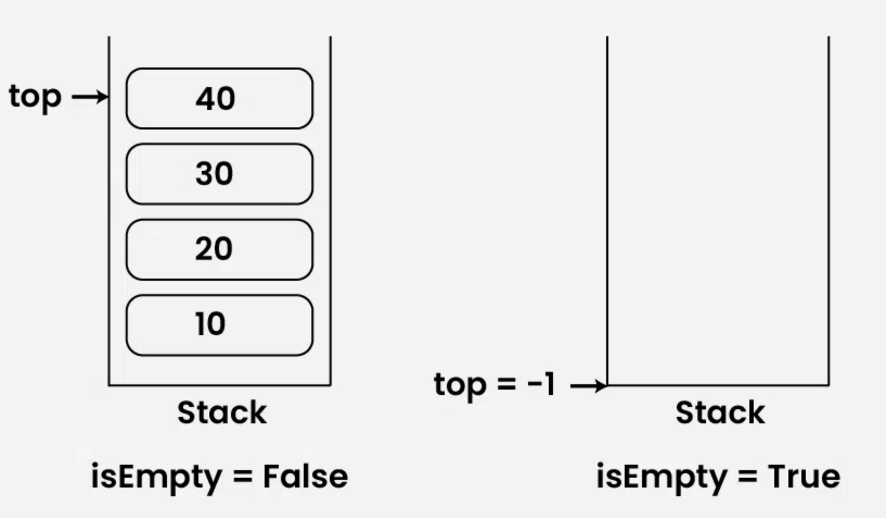


https://www.geeksforgeeks.org/dsa/introduction-to-stack-data-structure-and-algorithm-tutorials/

isEmpty Operation

Algorithm:

- Check for the value of **top** in stack.
- If **(top == -1)**, then the stack is **empty** so return **true** .
- Otherwise, the stack is not empty so return **false** .

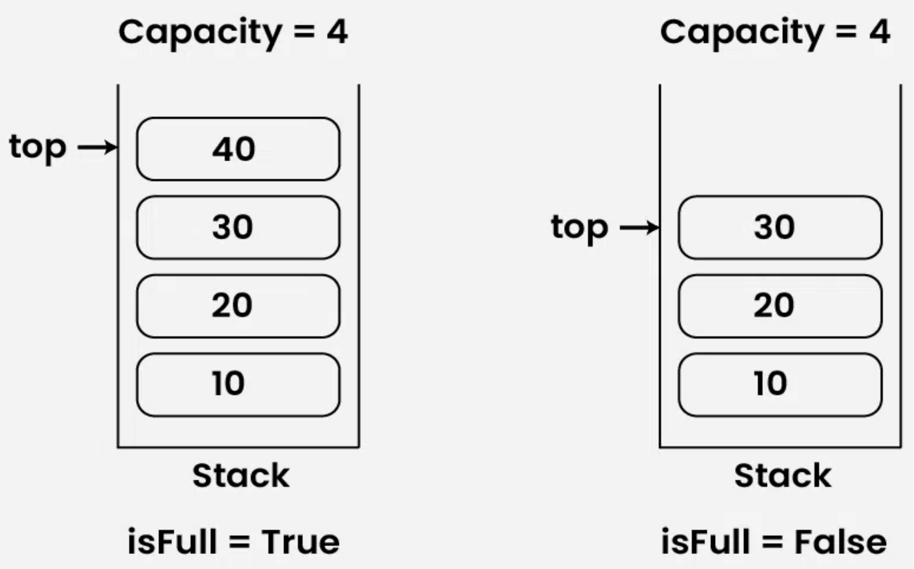


https://www.geeksforgeeks.org/dsa/introduction-to-stack-data-structure-and-algorithm-tutorials/

isFull Operation

Algorithm:

- Check for the value of top in stack.
- If (top == capacity-1), then the stack is full so return true.
- Otherwise, the stack is not full so return false.



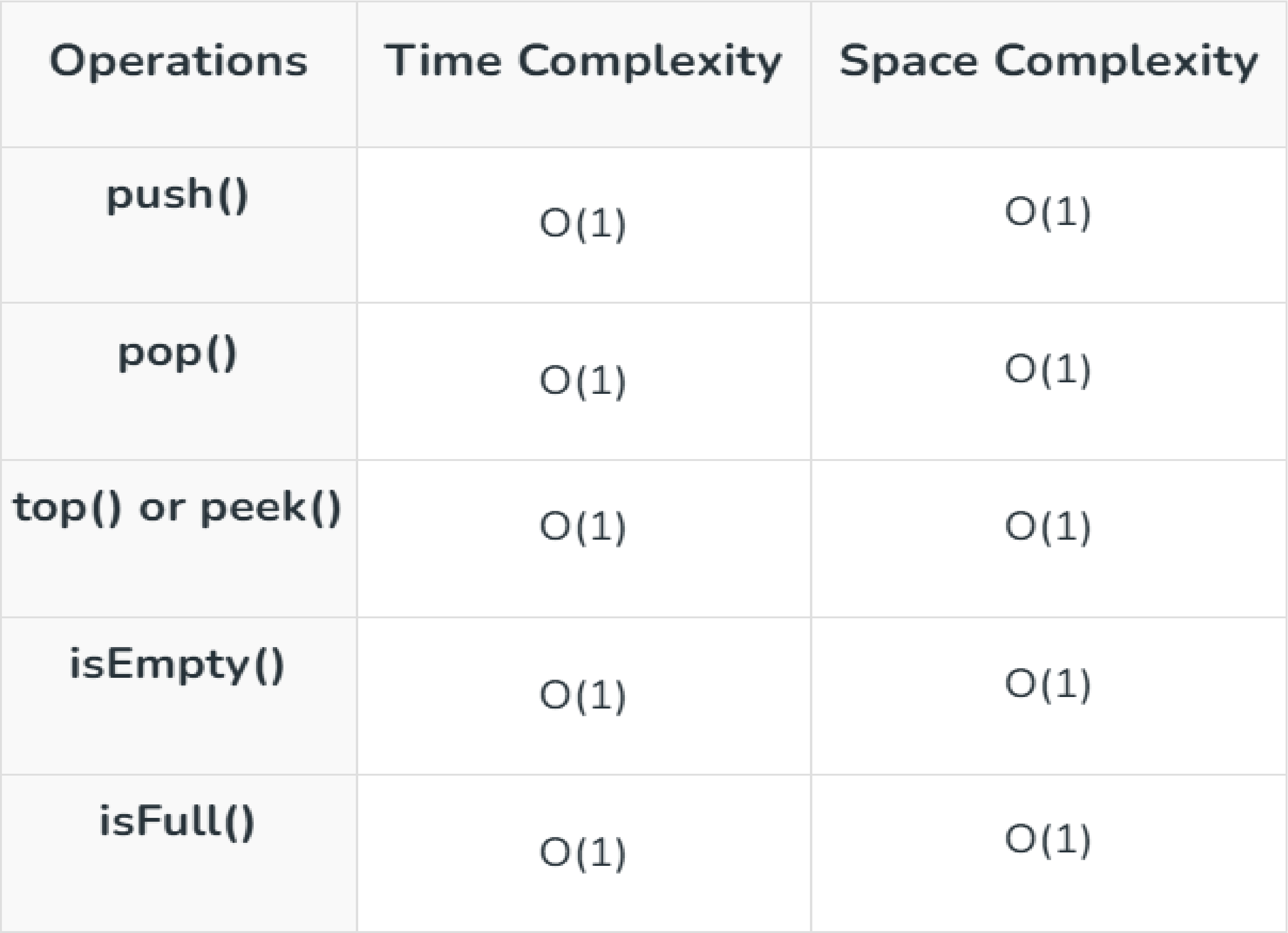
https://www.geeksforgeeks.org/dsa/introduction-to-stack-data-structure-and-algorithm-tutorials/

Code for push & pop operation

```
void newstack(stack <int> ss)
{
    stack <int> sg = ss;
    while (!sg.empty())
    {
        cout << '\t' << sg.top();
        sg.pop();
    }
    cout << '\n';
}
```

Complete code here

Complexity Analysis of Operations on Stack Data Structure



Operations	Time Complexity	Space Complexity
push()	O(1)	O(1)
pop()	O(1)	O(1)
top() or peek()	O(1)	O(1)
isEmpty()	O(1)	O(1)
isFull()	O(1)	O(1)

Test Your self

1. Which among the following represents a stack?

a: People waiting at a counter, where the action of popping is when someone has been served at the counter

b: A hand of bangles where pushing is wearing a new bangle

c: A pile of plates at a dinner party

d: People going around a merry-go-round

Test Your self

2. What is the time complexity of push operation in a stack? What is the time complexity of pop operation in a stack?

a: O(n), O(n)

b: O(1), O(n)

c: O(n), O(1)

d: O(1), O(1)

Test Your self

3. Consider these operations on an empty stack: push(3), push(5), pop(), push(10), push(11), pop(), push(100). What will be the stack configuration (first number is top of the stack, last is the bottom)

a: 100,10,3

b: 3,5,10,11,100

c: 3,10,100

d: None of the above

Test Your self

4. Consider a stack initially empty. Perform the following operations:

a) Push 5, Push 10, Push 15

b) Pop an element

c) Push 20

What is the final state of the stack?

Test Your self

Answers:

1. Correct Option: C
2. Correct Option : d
3. Correct Option :A

Review

1. **Definition**: Stack is a Last In, First Out (LIFO) data structure.
2. **Operations**: Push (adds to top), Pop (removes from top), Peek (returns top without removal), Is Empty, Size.
3. **Implementation**: Can be implemented using arrays, linked lists, or dynamic arrays (vectors in C++).
4. **Applications**: Used in expression evaluation, function call management, backtracking, undo mechanisms, parsing, browser history, text editors.
5. **Complexity**: O(1) time complexity for push, pop, peek, is Empty, and size operations.

THANK
YOU