

# DATA STRUCTURE

□ INTRODUCTION TO STACK

EMAN T. MAHDI  
COLLEGE OF C.S. & I.T.

# OVER VIEW

- STACK
- **INITIALIZE STACK**
- **STACK OPERATIONS**
- **STACK ALGORITHMS**



# STACK

A STACK IS AN ORDERED COLLECTION OF ITEMS (ELEMENTS), IN WHICH THE REMOVAL AND ADDITION OF STACK ITEMS CAN TAKE PLACE ONLY AT THE TOP OF THE STACK.

- TO ADD (PUSH) AN ITEM TO THE STACK, IT MUST BE PLACED ON THE TOP OF THE STACK.
- TO REMOVE (POP) AN ITEM FROM THE STACK, IT MUST BE REMOVED FROM THE TOP OF THE STACK TOO.

THUS, THE LAST ELEMENT THAT IS PUSHED INTO THE STACK, IS THE FIRST ELEMENT TO BE POPPED OUT OF THE STACK. I.E., LAST IN FIRST OUT (LIFO)

# OPERATIONS OF STACK

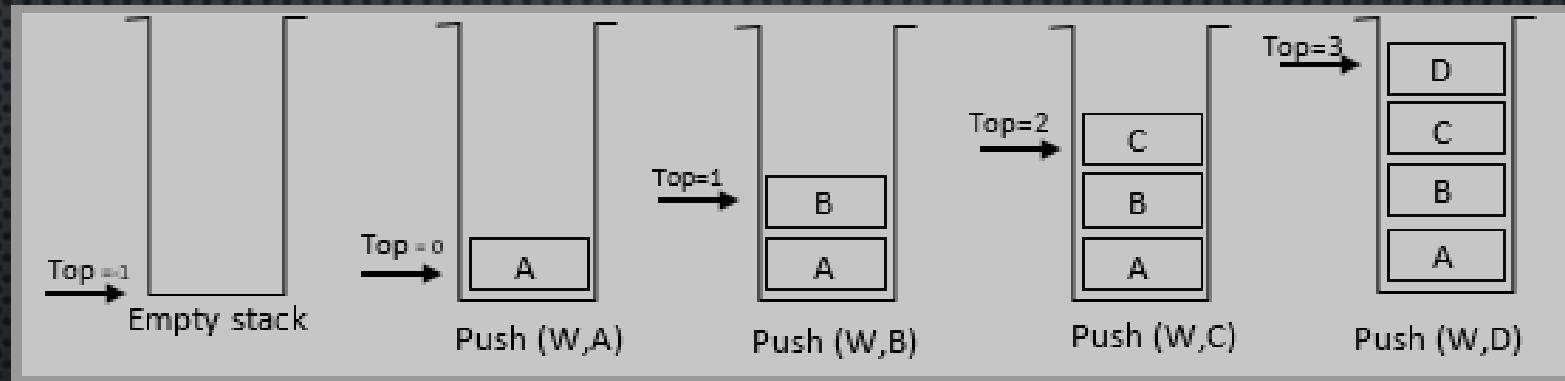
THE TWO MAIN OPERATIONS WHICH CAN BE APPLIED TO A STACK ARE GIVEN SPATIAL NAMES

**PUSH:** ADD AN ELEMENT TO THE TOP OF STACK

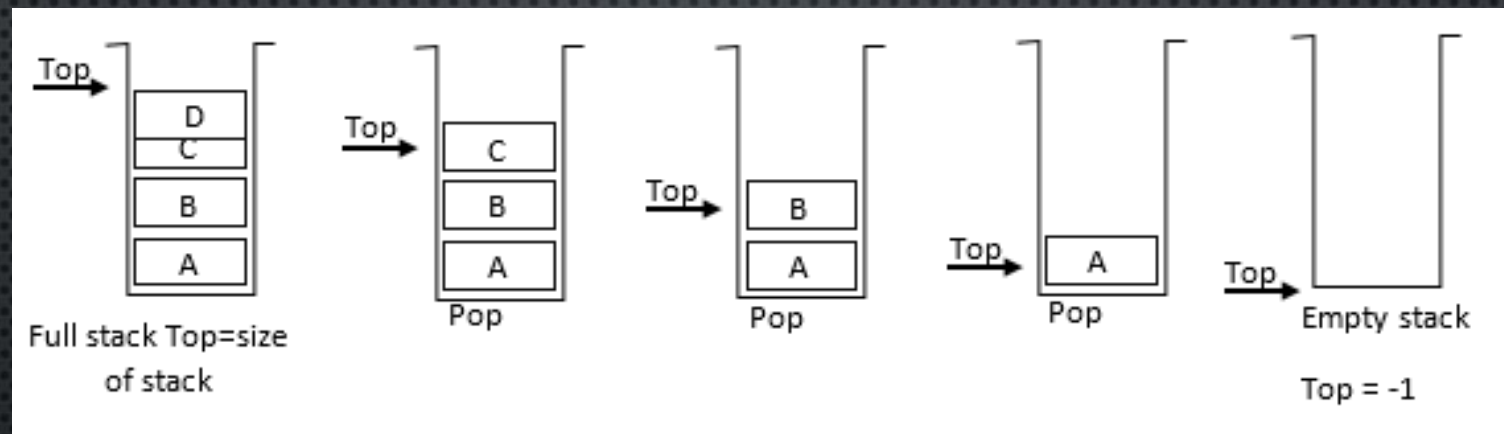
**POP:** DELETE THE ELEMENT AT THE TOP OF STACK



EXAMPLE: PERFORMING PUSH OPERATION WITH ITEM ABCD WITH A STACK W.



# POP OPERATION (DELETE ITEM FROM THE STACK)







# .TYPES OF STACK

- ARRAYS (STATIC: THE SIZE OF STACK IS GIVEN INITIALLY)
- LINKED LISTS (DYNAMIC: NEVER BECOME FULL)



# STACK ALGORITHMS

There are two algorithms to add and remove items into and from stack.

- PUSH ALGORITHM

- 1-[over flow]

- If  $Top \geq N$

- Then over flow

- 2-[increment pointer ]

- $Top = Top + 1$

- 3-[insert element]

- $Stack[Top] = \text{New element}$

# POP ALGORITHM

1-[under flow]

    If  $Top \leq -1$

        Then under flow

2-[unstack element ]

$element = Stack[Top]$

3-[decrement pointer]

$Top = Top - 1$

Notes: Top indicate to the top of stack which initialized (top =-1)



# REFERENCES

- : INTRODUCTION TO ALGORITHMS, 3RD EDITION BY THOMAS H. CORMEN ,CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN
- INTRODUCTION TO ALGORITHMS, 3RD EDITION BY THOMAS H. CORMEN ,CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN
- ELEMENTS OF PROGRAMMING INTERVIEWS IN JAVA: THE INSIDERS' GUIDE, BY ADNAN AZIZ, TSUNG-HSIEN LEE, AMIT PRAKASH
- [HTTPS://GITHUB.COM/CAREERMONK/DATASTRUCTURESANDALGORITHMSMADEEASY](https://github.com/careermonk/DataStructuresAndAlgorithmsMadeEasy)