

Data Structures and Algorithms

MSc. KhangVQH

Faculty Of Information Technology

Fall - 2022

Lecture #4

STACK



Terminology: Push ,Pop, display ,Top

content

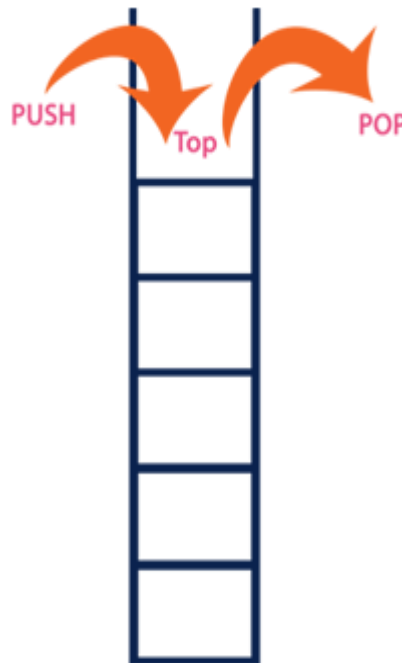
- What is Stack?
- Stack working principle

What is a Stack?

- **Stack** is a **linear data structure** in which the **insertion** and **deletion** operations are performed **at only one end**.
- In a stack, **adding** and **removing** of elements are performed **at a single position** which is known as **"top"**.
- That means, a **new element** is **added at top** of the **stack** and an element is **removed from the top of the stack**.

Principle

LIFO(Last In First Out)



Stack Examples



top

4

50

3

40

2

30

1

20

0

10

CD-BOX



IDLY PLATES



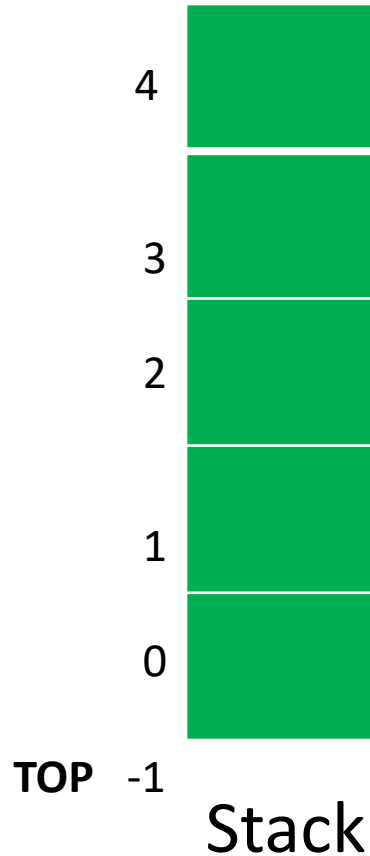
Operations on a Stack

- **Push:** đẩy 1 phần tử vào đầu ngăn xếp.
- **Pop:** xóa 1 phần tử đầu khỏi ngăn xếp.
- **Display:** hiển thị các phần tử trong stack
- **peek:** lấy 1 phần tử đầu của ngăn xếp mà không xóa phần tử đó.
- **isFull:** kiểm tra ngăn xếp đã đầy chưa.
- **isEmpty:** kiểm tra ngăn xếp có rỗng không.

Push (To insert an element on to the stack)

Initial Top=-1

SIZE=5



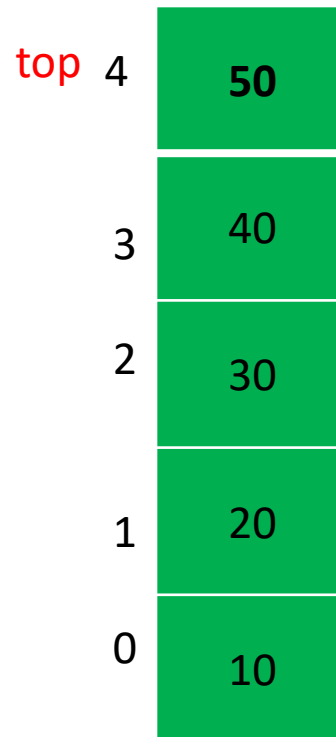
Push (To insert an element on to the stack) Algorithm

- **Step 1** - Check whether **stack** is **FULL**.
(**top == SIZE-1**)
- **Step 2** - If it is **FULL**, then display
"**Stack is FULL!!! Insertion is not possible!!!**" and terminate the function.

Push()

Push(60)

TOP=4 ,i.e. Stack is FULL

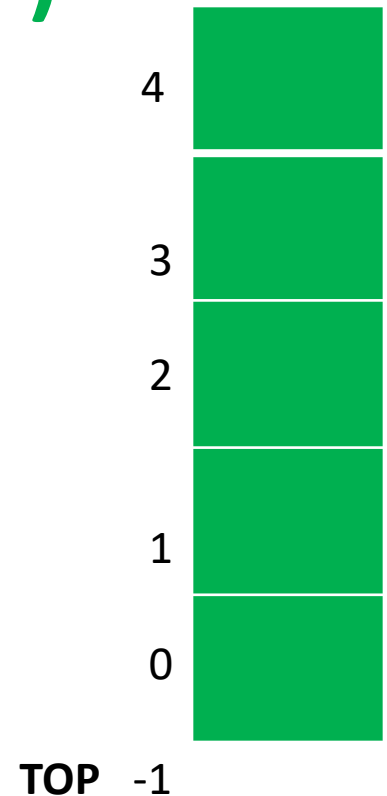


stack

Push()

- **Step 3** - If it is **NOT FULL**, then increment **top** value by one (**top++**) and set `stack[top]` to value (**`stack[top] = value`**).

❖ $TOP = -1$,i.e. Stack is empty



Push (To insert an element on to the stack)

If (top == SIZE-1)

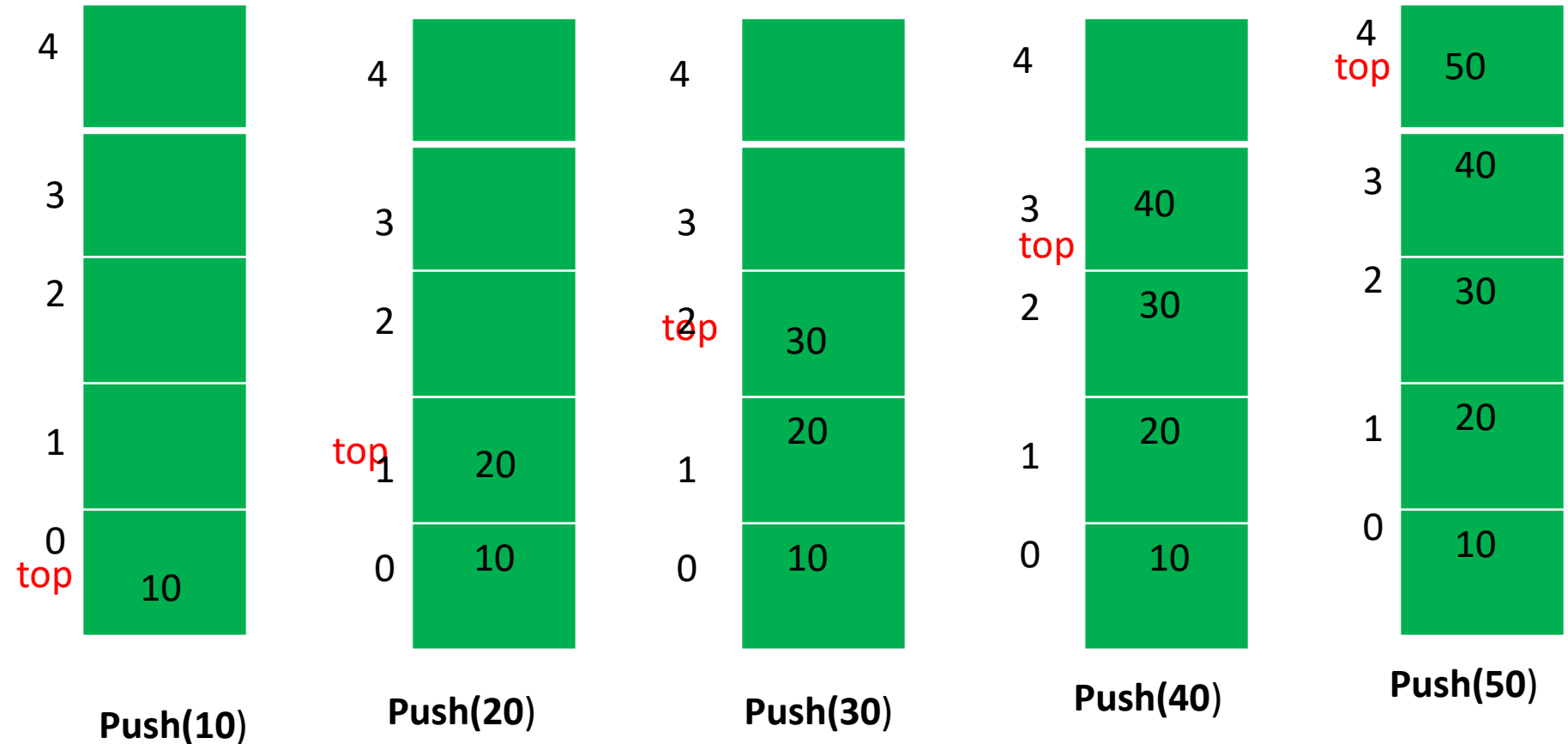
-1==4

0==4

1==4

2==4

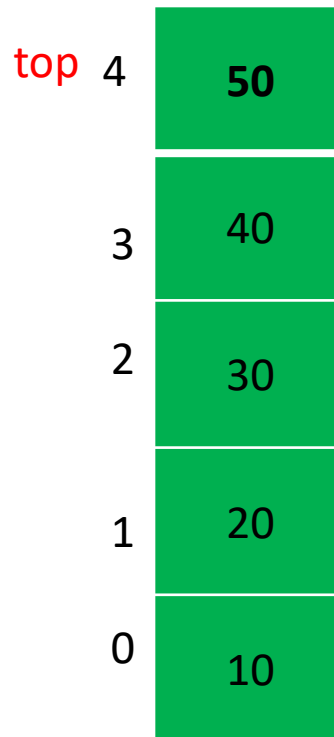
3==4



Push()

Push(60) **If (top == SIZE-1)**

TOP=4 ,i.e. Stack is FULL Insertion not possible

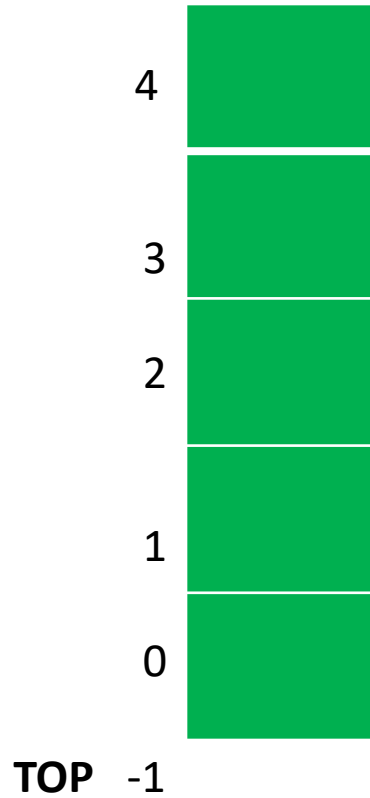


Pop (To delete an element from the stack) Algorithm

- **Step 1** - Check whether **stack** is **EMPTY**.
(**top == -1**)
- **Step 2** - If it is **EMPTY**, then display "**Stack is EMPTY!!! Deletion is not possible!!!**" and terminate the function.

Pop()

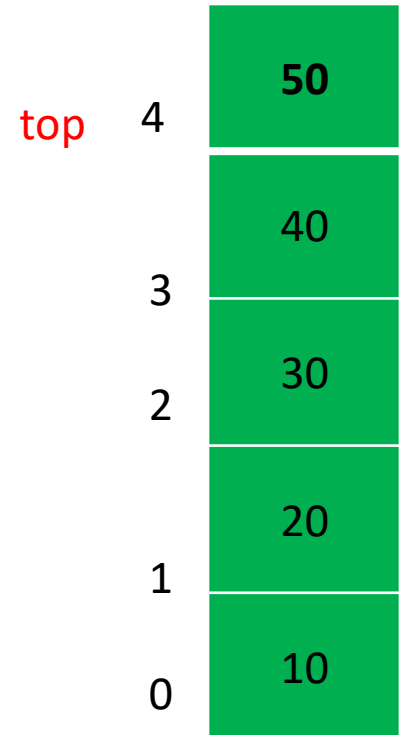
TOP=-1 ,i.e. Stack is empty



Pop()

TOP=4 ,i.e. Stack is not empty

Step 3 - If it is **NOT EMPTY**, then delete **stack[top]** and decrement **top** value by one (**top--**).



Pop (To delete an element from the stack)

If (top == -1)

4== -1

3== -1

2== -1

1== -1

0== -1

4
top

4

top

4

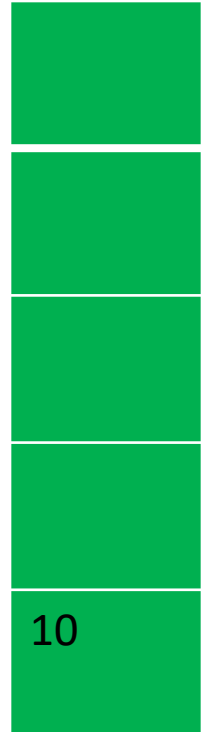
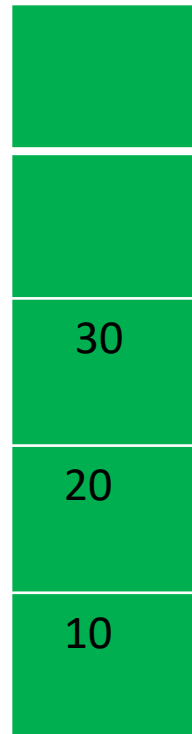
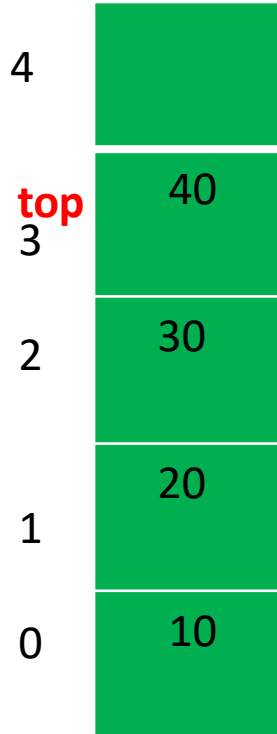
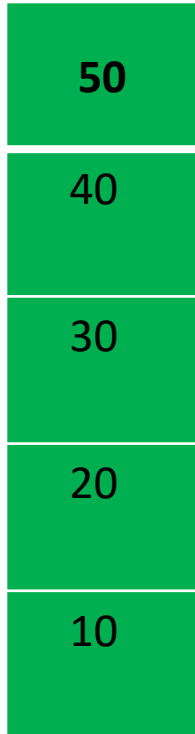
top

4

top

4

top



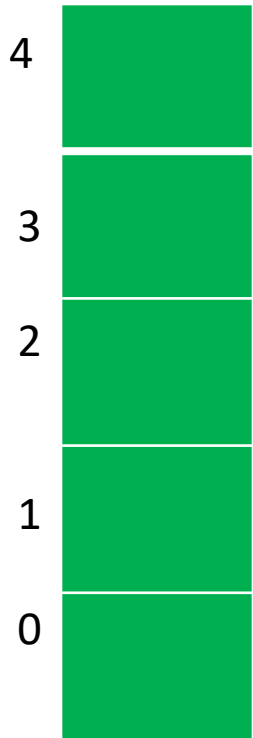
Pop(50)

Pop(40)

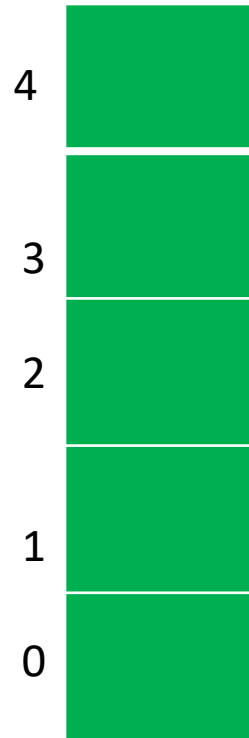
Pop(30)

Pop(20)

Pop (To delete an element from the stack)



Top=-1
Pop(10)



Stack is empty

TOP -1

Display (To display elements of the stack)

- **Step 1** - Check whether **stack** is **EMPTY**. (**top == -1**)
- **Step 2** - If it is **EMPTY**, then display "**Stack is EMPTY!!!**" and terminate the function.
- **Step 3** - If it is **NOT EMPTY**, then define a variable '**i**' and initialize with **top**. Display **stack[i]** value and decrement **i** value by one (**i--**).
- **Step 4** - Repeat above step until **i** value becomes '**0**'.

Display()

top 4	50
3	40
2	30
1	20
0	10

Implementation of Stack using Array

```
#include<stdio.h>
#include<conio.h>

#define SIZE 10

void push(int);
void pop();
void display();

int stack[SIZE], top = -1;

void main()
{
    int value, choice;
    clrscr();
    while(1){
        printf("\n\n***** MENU *****\n");
        printf("1. Push\n2. Pop\n3. Display\n4. Exit");
        printf("\nEnter your choice: ");
        scanf("%d",&choice);
        switch(choice){
            case 1: printf("Enter the value to be insert: ");
                    scanf("%d",&value);
                    push(value);
                    break;
            case 2: pop();
                    break;
            case 3: display();
                    break;
            case 4: exit(0);
            default: printf("\nWrong selection!!! Try again!!!");
        }
    }
}
```

```
void push(int value){
    if(top == SIZE-1)
        printf("\nStack is Full!!! Insertion is not possible!!!");
    else{
        top++;
        stack[top] = value;
        printf("\nInsertion success!!!");
    }
}

void pop(){
    if(top == -1)
        printf("\nStack is Empty!!! Deletion is not possible!!!");
    else{
        printf("\nDeleted : %d", stack[top]);
        top--;
    }
}

void display(){
    if(top == -1)
        printf("\nStack is Empty!!!");
    else{
        int i;
        printf("\nStack elements are:\n");
        for(i=top; i>=0; i--)
            printf("%d\n",stack[i]);
    }
}
```

3. Display

4. Exit

Enter your choice: 1

Enter the value to be insert: 20

Insertion success!!!

***** MENU *****

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: 3

Stack elements are:

20

10

***** MENU *****

1. Push

2. Pop

3. Display

4. Exit

Enter your choice: