CS & IT ENGINEERING

Data Structures

Stack and Queues

Lecture No.- 01

By- Pankaj Sharma Sir

Recap of Previous Lecture

Topic Linked List Part-06/07

- PYQs ✓
- PS
- Types of LL

Topics to be Covered

Topic Stack and Queues Part 01

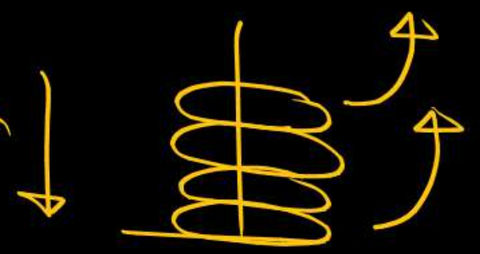
Topic : Stack - 1

DS

- Abstract view
  - (i) No coding
  - (ii) No implementation
  - (iii) No prog. language
  - (iv) feature/operations defined
- Concrete view
  - (i) Implementation
  - (ii) Prog. language

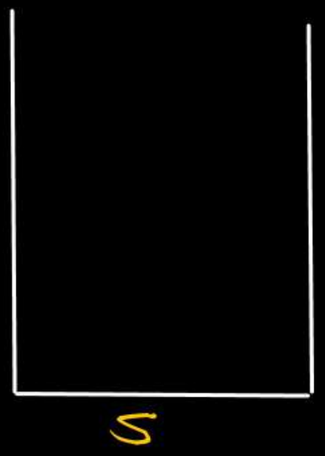
Stack

* Linear data structure.
* Order of deletion of element is opposite order of insertion.
* It works on Last-In First-Out policy.
* Elements are inserted & deleted only from one end called TOP of the Stack.



Stack as ADT

Stack of numbers



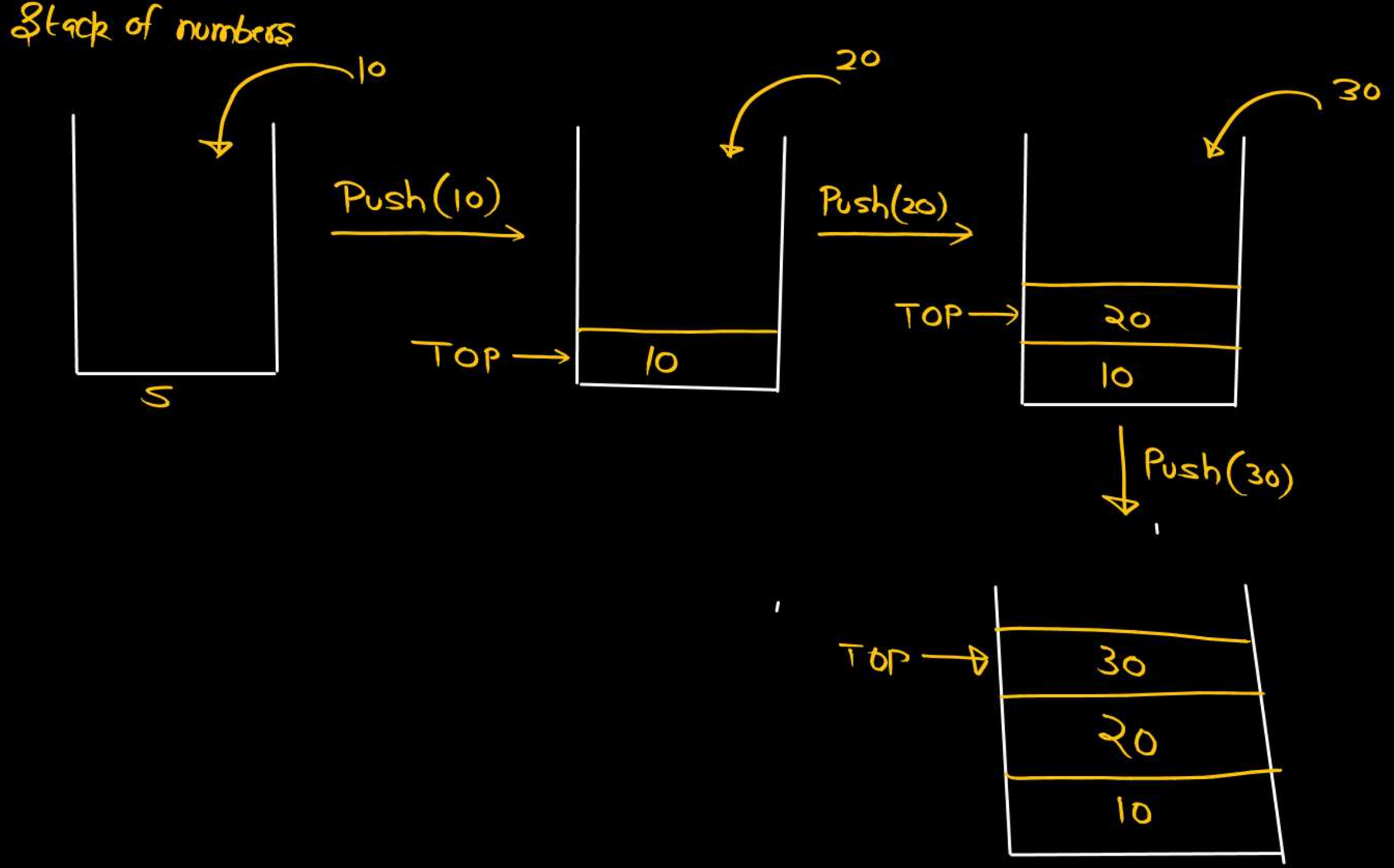
(i) Push() : Insert

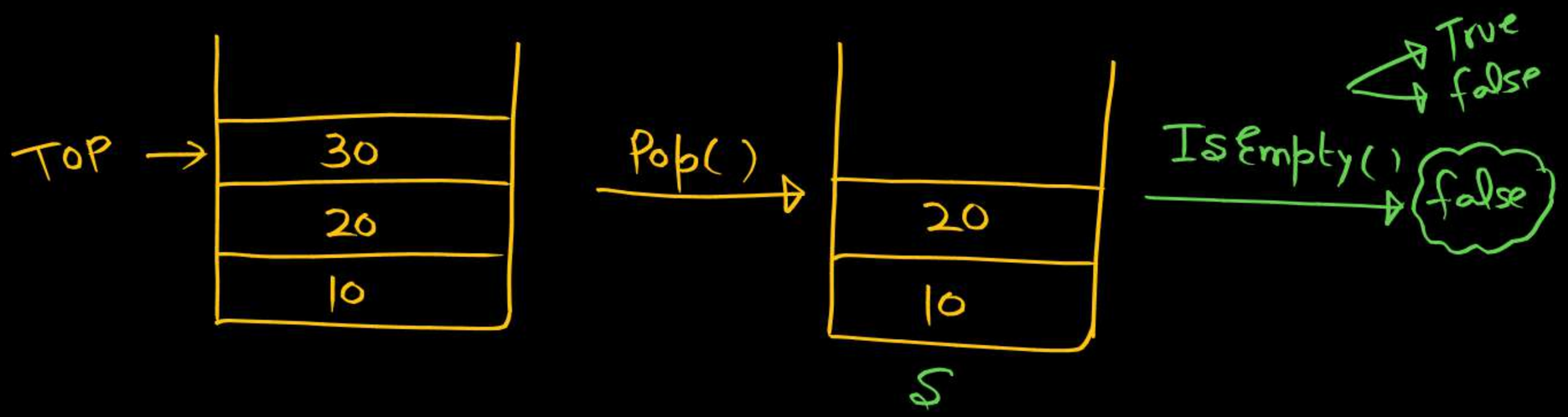
(ii) Pop() : delete

Initially: Empty stack

TOP : Points to most recently added element

Stack as ADT


Stack of numbers
10
20
30
Push(10)
Push(20)
TOP
10
TOP
20
10
S
Push(30)
TOP
30
20
10



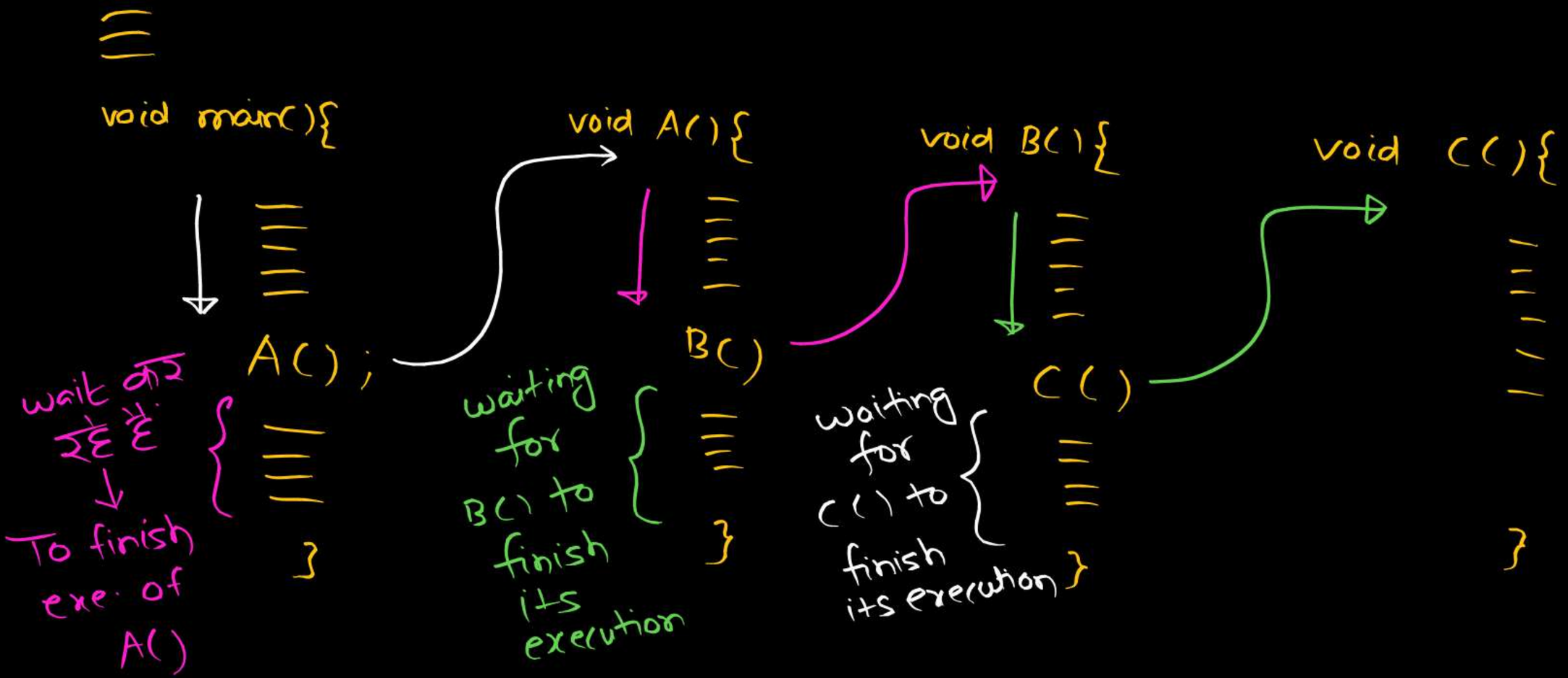
(i) Push
(ii) Pop
(iii) IsEmpty

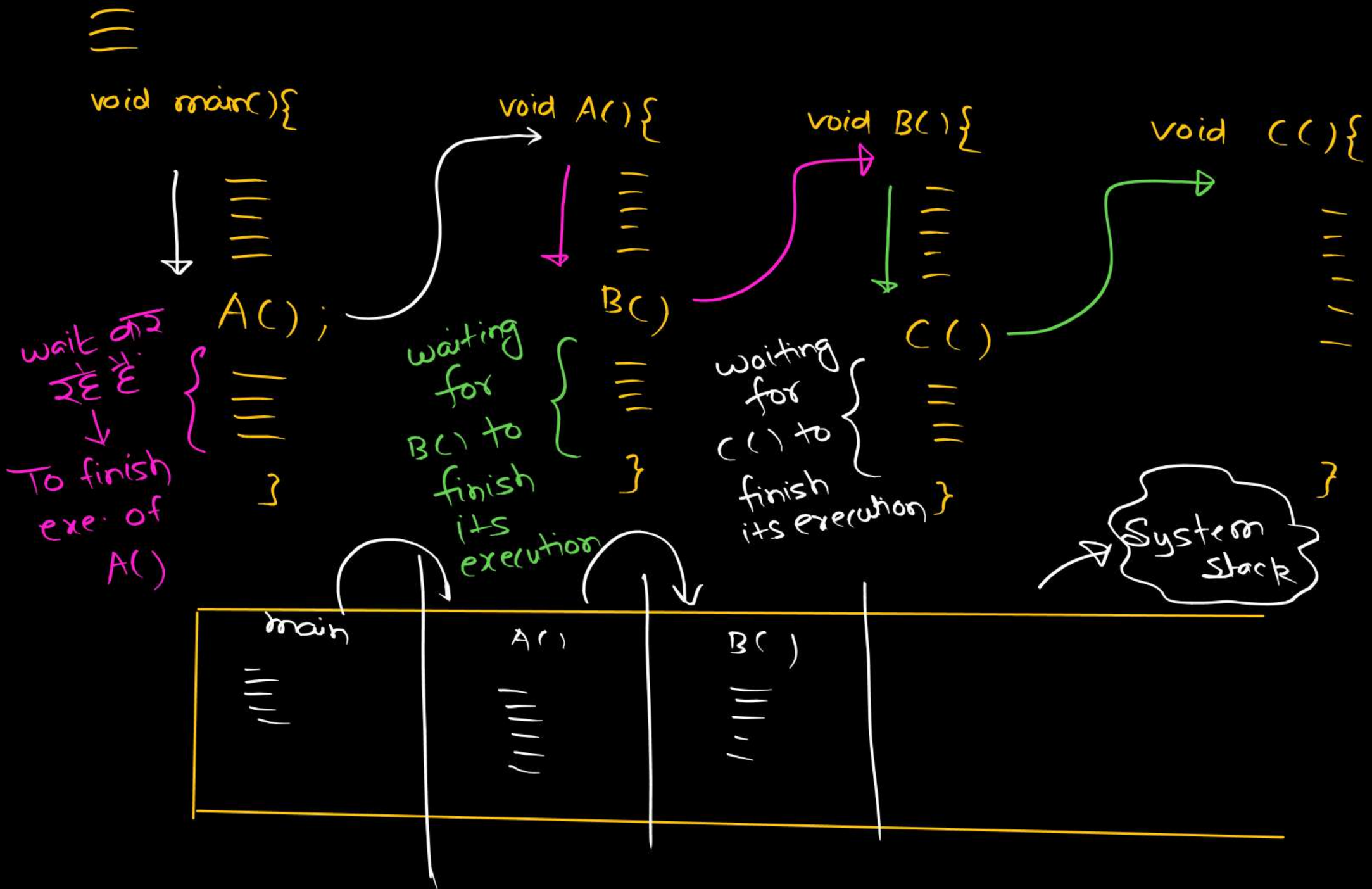
} IsFull( )

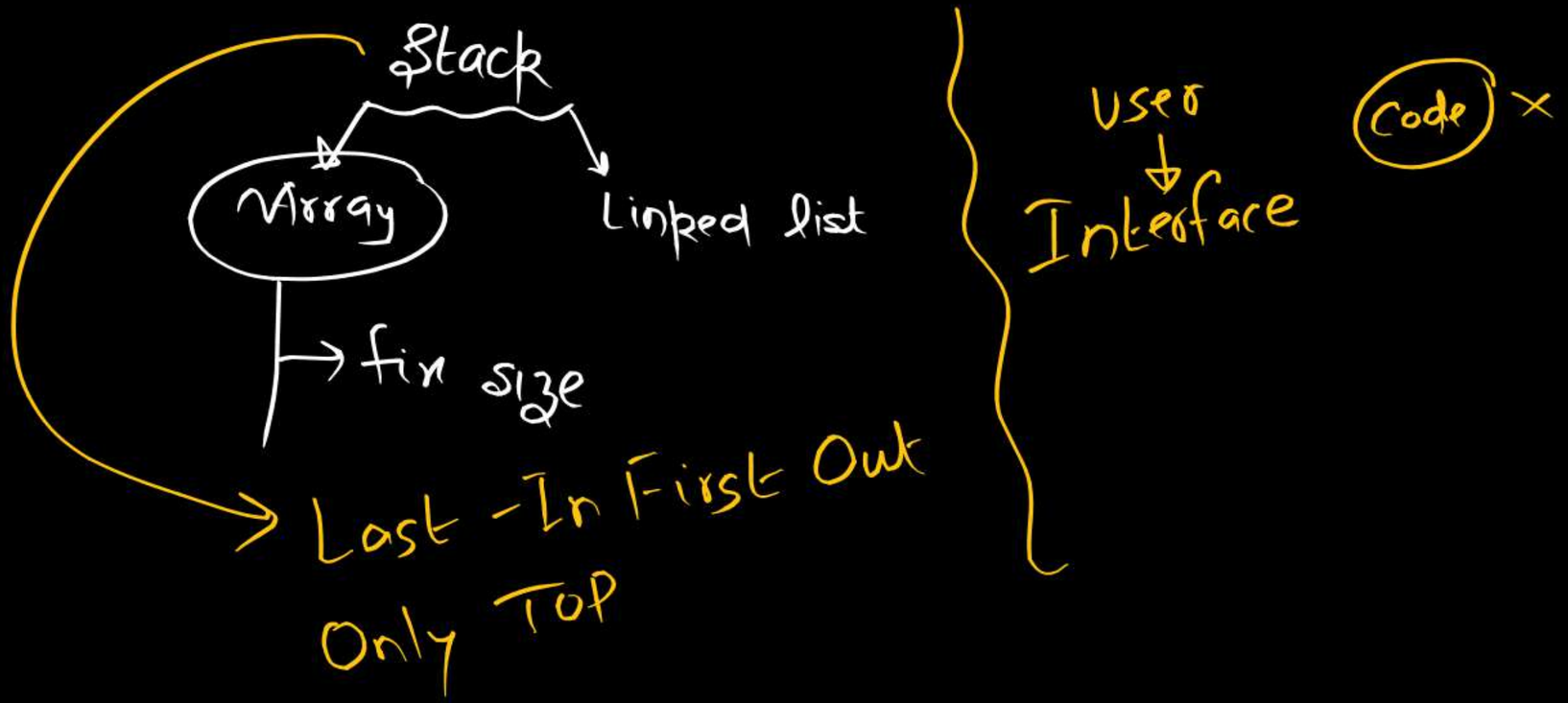
Applications

1. Tower of Hanoi
2. DFS
3. Backtracking
4. Recursion
5. Infix to prefix conversion
6. Infix to postfix conversion
7. prefix Evaluation
8. postfix Eval.
9. Balanced parenthesis checking

To delay
To post-poned certain decision
wait करवानी

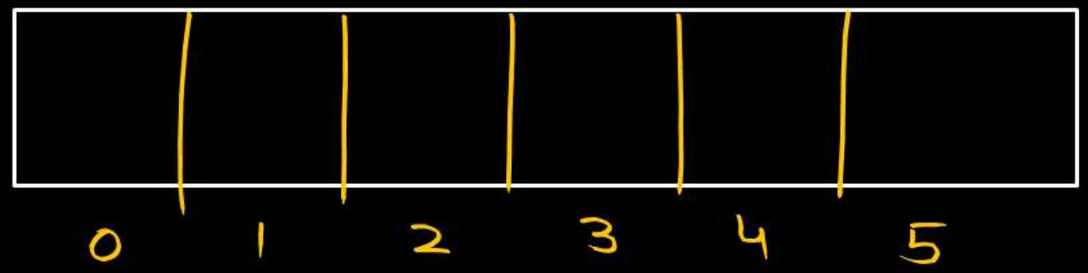

void main(){
A();
wait कर रहे हैं
→ To finish exe. of A()
}
void A(){
B()
waiting for B() to finish its execution
}
void B(){
C()
waiting for C() to finish its execution
}
void C(){
}


void main(){
A();
wait करेंगे
↓
To finish exe. of A()
}
void A(){
B()
waiting for B() to finish its execution
}
void B(){
C()
waiting for C() to finish its execution
}
void C(){
}
System Stack
main
A()
B()


Stack
Array
Linked list
fix size
Last - In First Out
Only TOP
User
Interface
Code
x

```
#define SIZE 6
int stack[SIZE];
```

stack



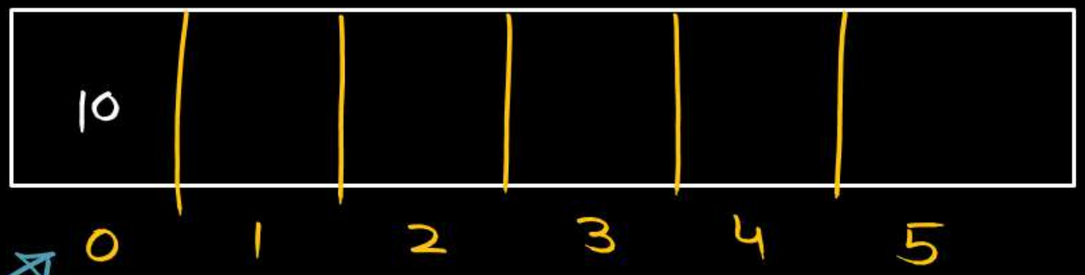
Initially, stack is Empty.

TOP : Most recently added elements

→ No valid index

TOP = -1

```
#define SIZE 6
int stack[SIZE];
```

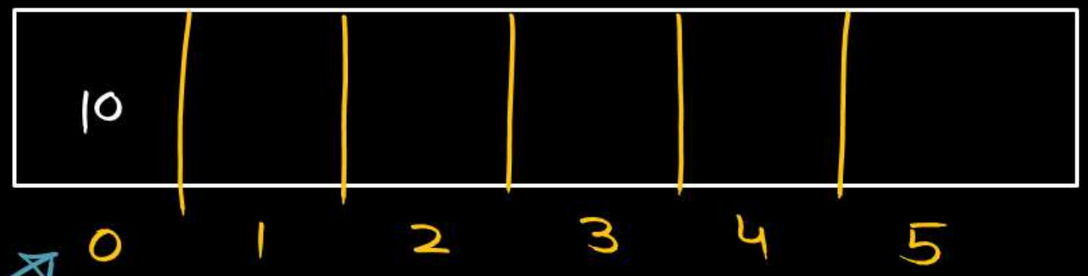


TOP = ~~-1~~

1.) Push(10) : TOP = TOP+1

stack[TOP] = ele

```
#define SIZE 6
int stack[SIZE];
int TOP = -1
```



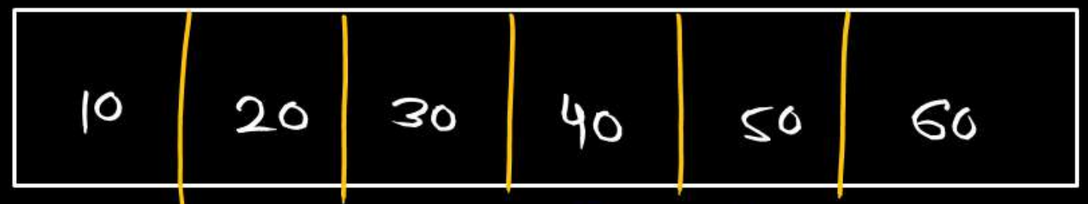
TOP = -1

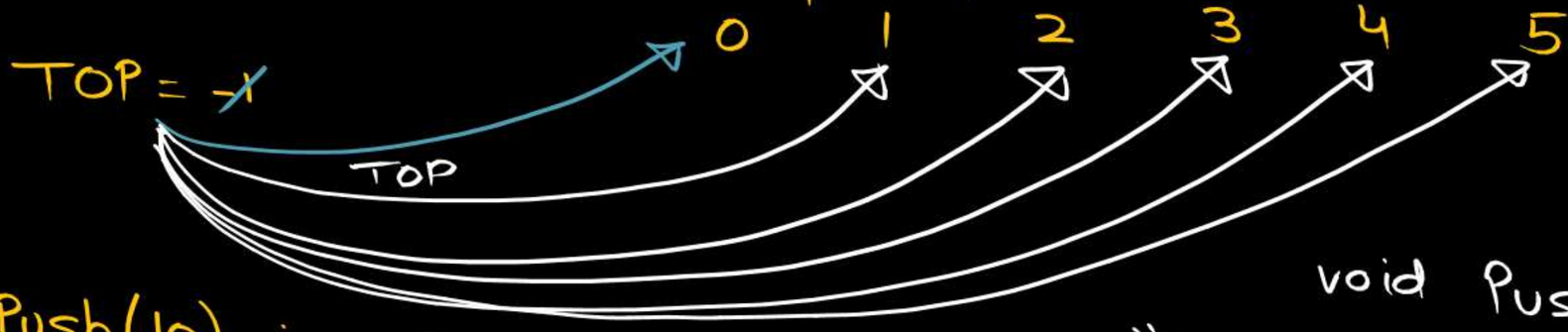
1.) Push(10) : TOP = TOP+1
stack[TOP] = ele

दोनों है

```
void Push( int x )
{
    TOP = TOP + 1
    Stack[TOP] = x;
}
```

```
#define SIZE 6
int stack[SIZE];
int TOP = -1
```





1.) Push(10) : 0

2.) Push(20) : 1

3.) Push(30) : 2

4.) Push(40) : 3

5.) Push(50) : 4

6.) Push(60) : 5

होता है

Push(70) : overflow

insert

```
void Push( int x )
{
    TOP = TOP + 1
    Stack[TOP] = x;
}
```

No loop
No rec.

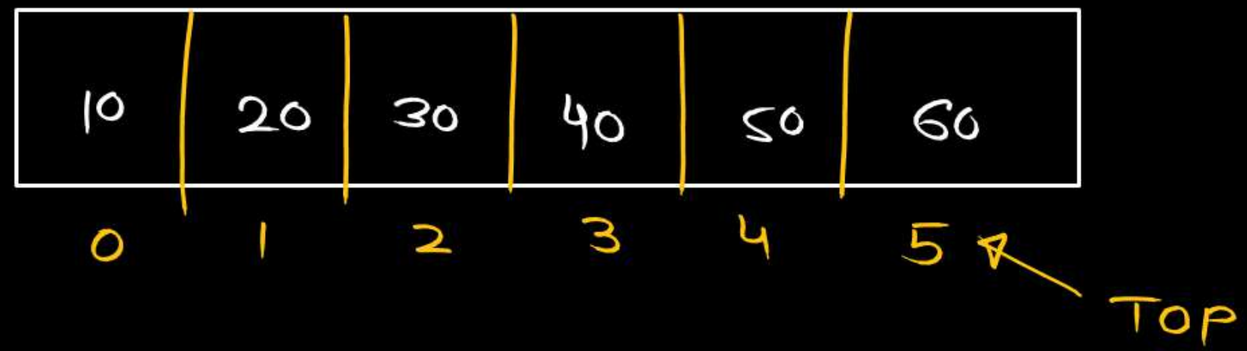
Constant time

O(1)

```
void Push(int x)
{
    if ( TOP == SIZE-1)
        {
          pf("overflow");
          return;
        }
    TOP = TOP+1
    stack[TOP] = x;

}
```

```
#define SIZE 6
int stack[SIZE];
int TOP= -1
```

stack

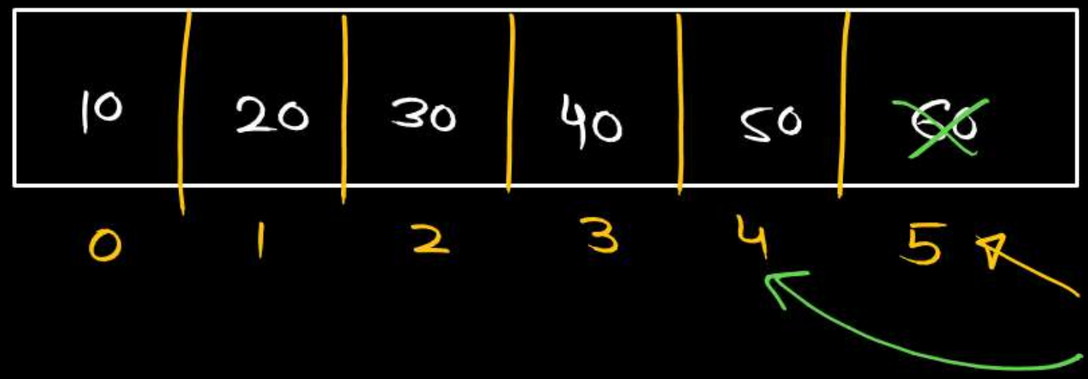


TOP = 5

```
int Pop( ) {
    int temp;
    temp = stack[TOP];
    TOP = TOP-1;

}
```

```
#define SIZE 6
int stack[SIZE];
int TOP = -1
```

stack

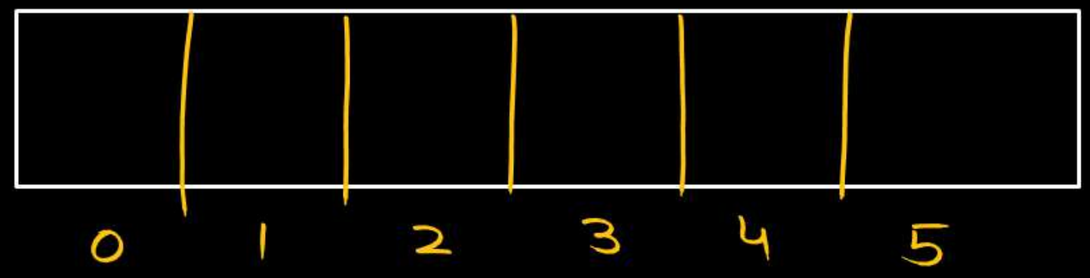


TOP = ~~5~~ 4

```
int Pop( ) {
    int temp;
    temp = stack[TOP];
    TOP = TOP-1;
    return temp;
}
```

डेटा है

```
#define SIZE 6
int stack[SIZE];
int TOP= -1
```



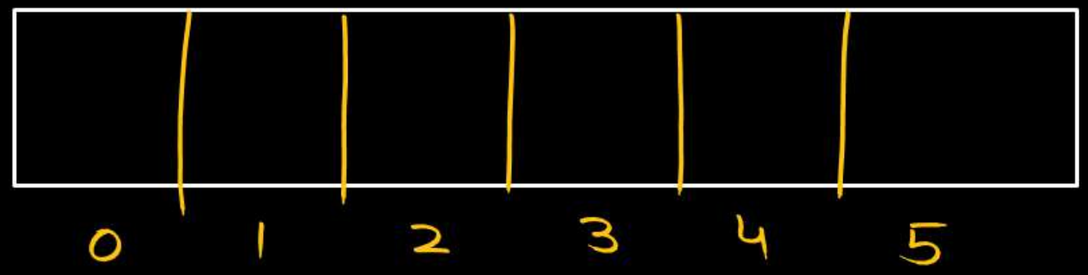
TOP = -1

Underflow

```
int Pop( ) {
    int temp;
    temp = stack[TOP];
    TOP = TOP-1;
    return temp;
}
```

```
#define SIZE 6
int stack[SIZE];
int TOP = -1
```

stack



TOP = -1

Problem }

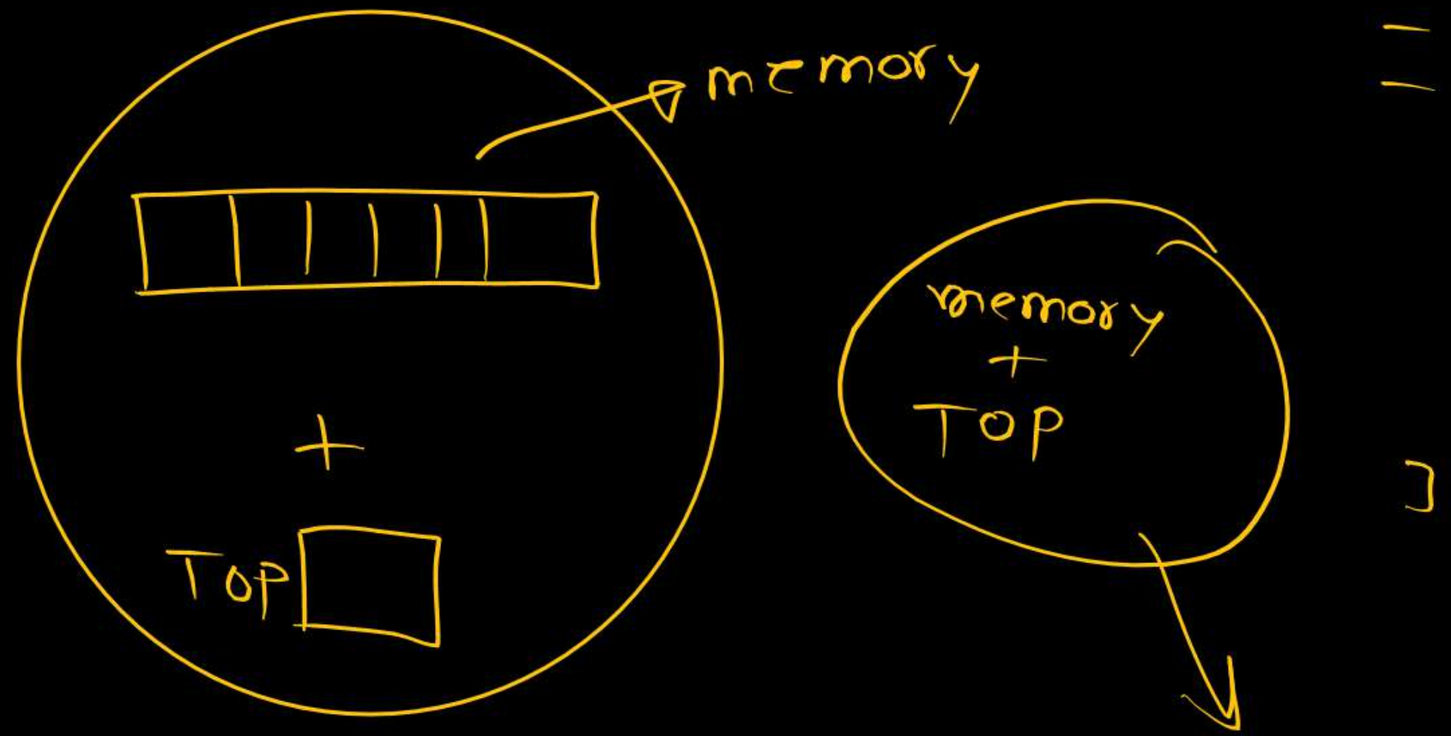
Underflow

int Pop( ) {

```
if(TOP==-1){
    return INT_MIN;
}
int temp;
temp = stack[TOP];
TOP = TOP-1;
return temp;
```

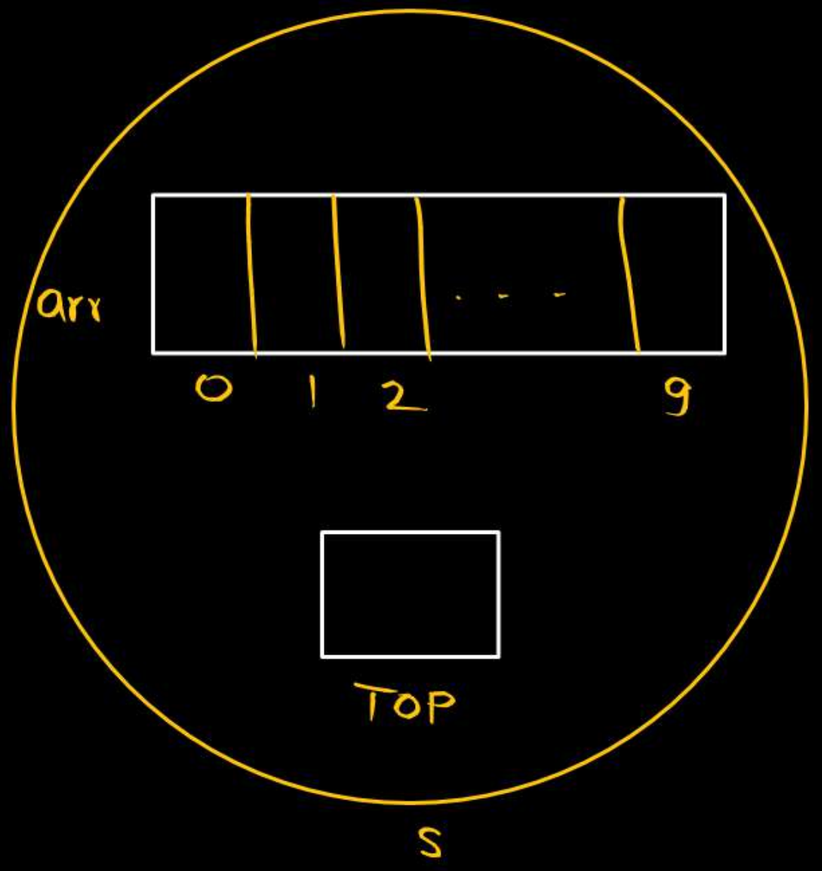
Constant time

Collection of → Memory → int stack[ ]; → int TOP = -1;
→ TOP → int TOP; → void main() {



```
#define SIZE 10
struct stack {
        int arr[SIZE];
        int TOP;
            }

void main(){
        struct stack s;
        s.TOP=-1;
            =
            =
        }
```

No memory is allocate



```
#define SIZE 10
struct stack {
        int arr[SIZE];
        int TOP;
            }

void main(){
        struct stack S1,S2;
        S.TOP=-1;
            =
            =
        }
```

No memory is allocate

Push( )

(i) कौन से stack में

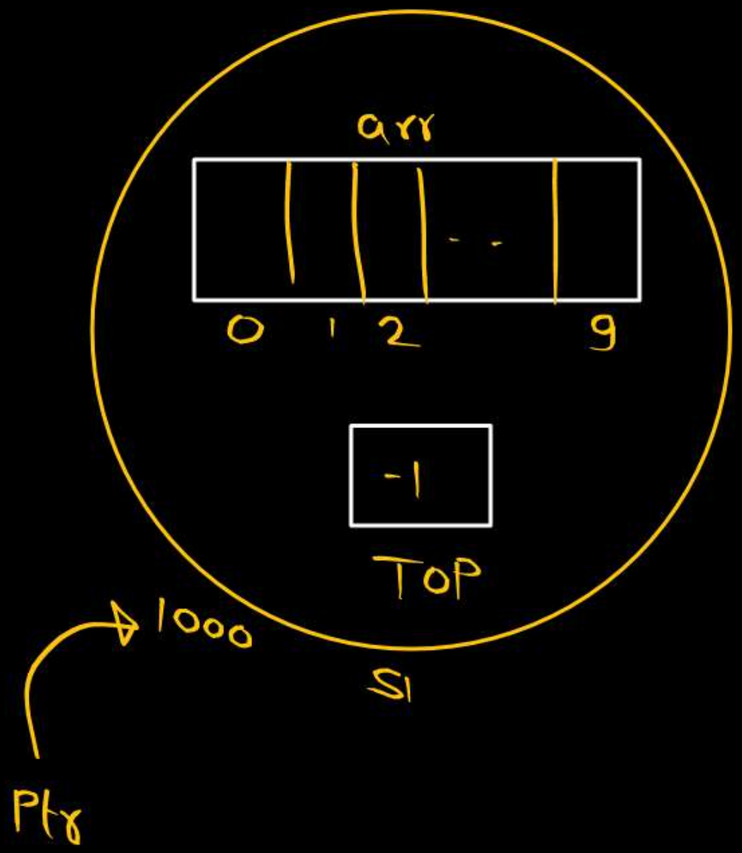
(ii) What to push

Call by value

```
Push(S1,10);
Push(S2,20);
```

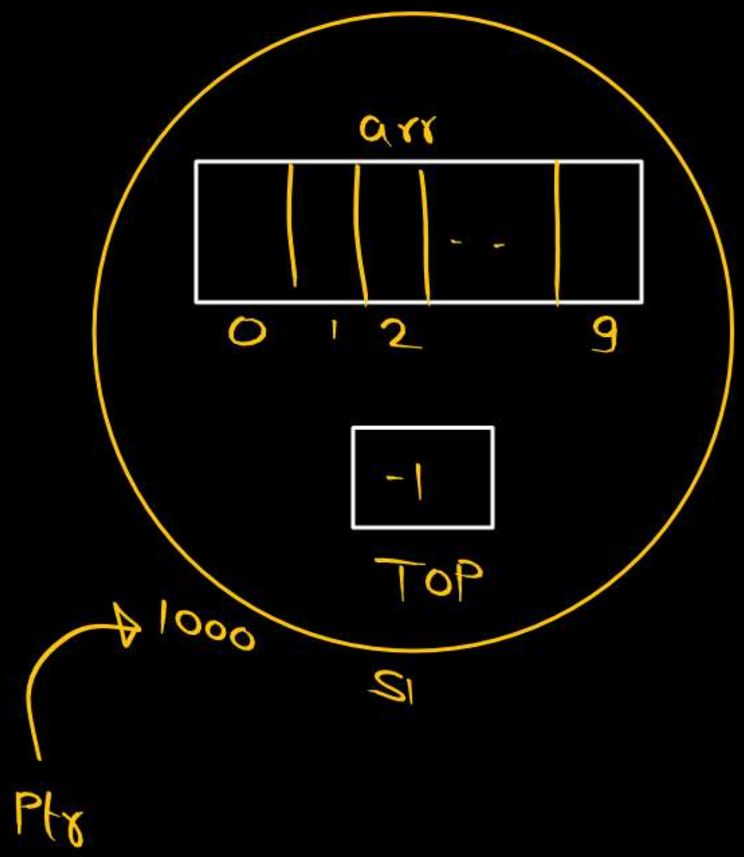
```
Push(&s1, 10);
Push(&s2, 20);
```

```
Struct stack{
        int arr[SIZE];
        int TOP; };
void main(){

struct stack S1, S2;
 S1.TOP = S2.TOP = -1;

 Push(&S1, 10);
```




```
void Push(struct stack *Ptr, int x)
    {
        if ( TOP == SIZE-1)
            ⇓
        if( Ptr → TOP == SIZE-1)
            {



            }

    }
```

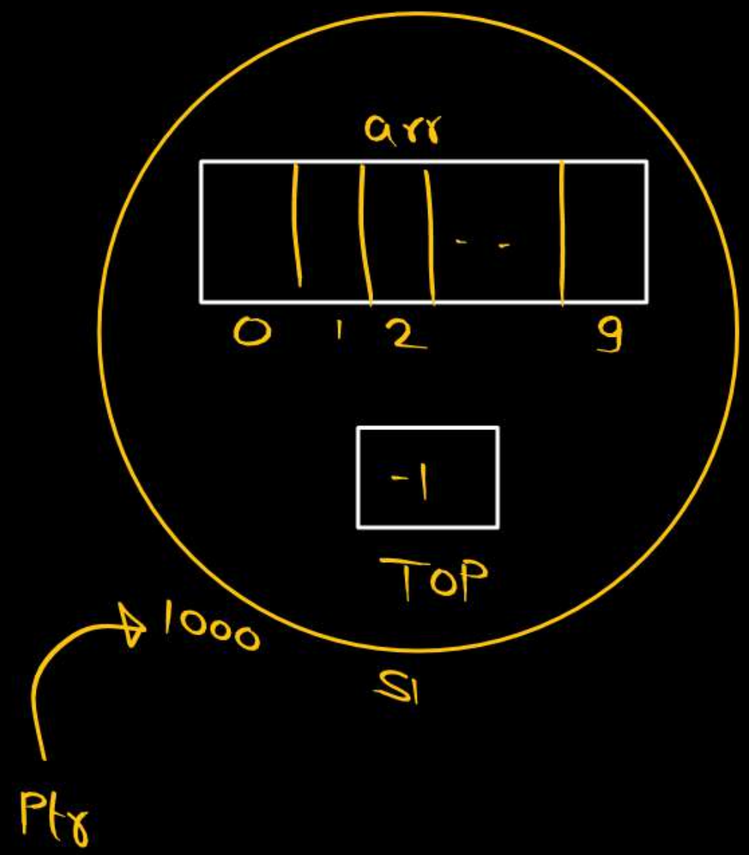
```
Struct stack{
        int arr[SIZE];
        int TOP; };
void main(){

struct stack S1, S2;
 S1.TOP = S2.TOP = -1;

 Push(&S1, 10);
```




```
void Push(struct stack *Ptr, int x)
    {
      if( Ptr->TOP == SIZE-1 )
            return;
      TOP = TOP + 1;   ✗
Ptr->TOP = Ptr->TOP + 1;   ✓
    }
```

```
Struct stack{
        int arr[SIZE];
        int TOP; };
void main(){

struct stack S1, S2;
 S1.TOP = S2.TOP = -1;

 Push(&S1, 10);
```

arr[TOP] = x




```
void Push(struct stack *Ptr, int x)
    {
      if( Ptr->TOP == SIZE-1)
            return;
      Ptr->TOP = Ptr->TOP +1

      Ptr->arr[Ptr->TOP] = x;


    }
```

Stack permutation

Given $n = 3$

1, 2, 3

no. of permutations

$3!$

1) 1, 2, 3
2) 1, 3, 2
3) 2, 1, 3
4) 2, 3, 1
5) 3, 1, 2
6) 3, 2, 1

6 possible permutation

Given 3 elements 1, 2, 3 and their order of insertion is 1, 2, 3 (fix)

u can perform pop() anytime.

Q) Possible order of pop()

GATE

THANK - YOU