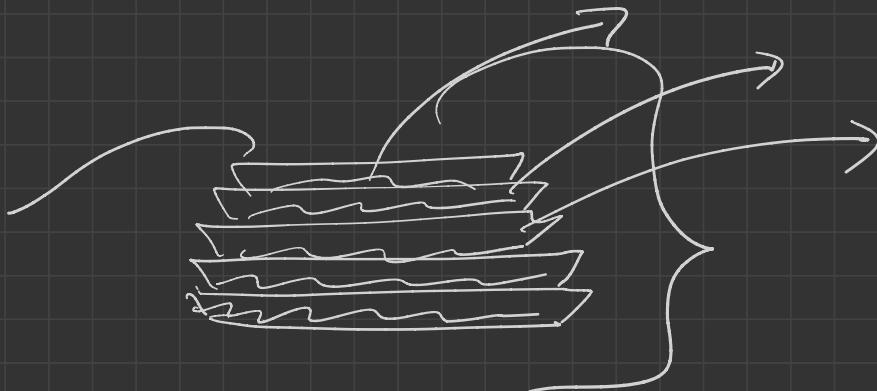



Stack

→ what - ?

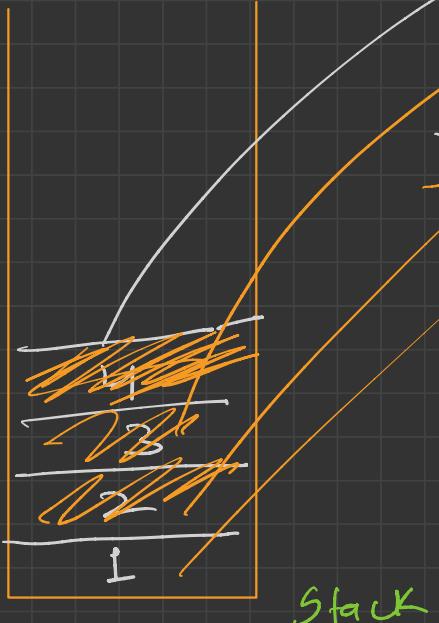


\rightarrow LIFO \rightarrow Last In first Out

Stack \rightarrow D.S

insert
Push \rightarrow operation
1, 2, 3, 4

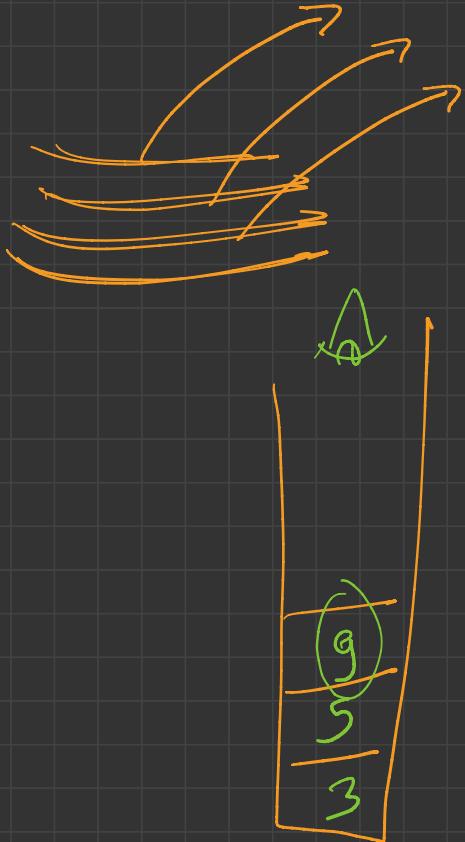
1, 2, 3, 4



4, 3, 2, 1
remove (pop operation)

LIFO order

Stack :- D.S \rightarrow LIFO order



Operations :-

- ↳ push
- ↳ pop
- ↳ peek \rightarrow top element
- ↳ isEmpty empty

STL

Creation \rightarrow

stack < int > s;

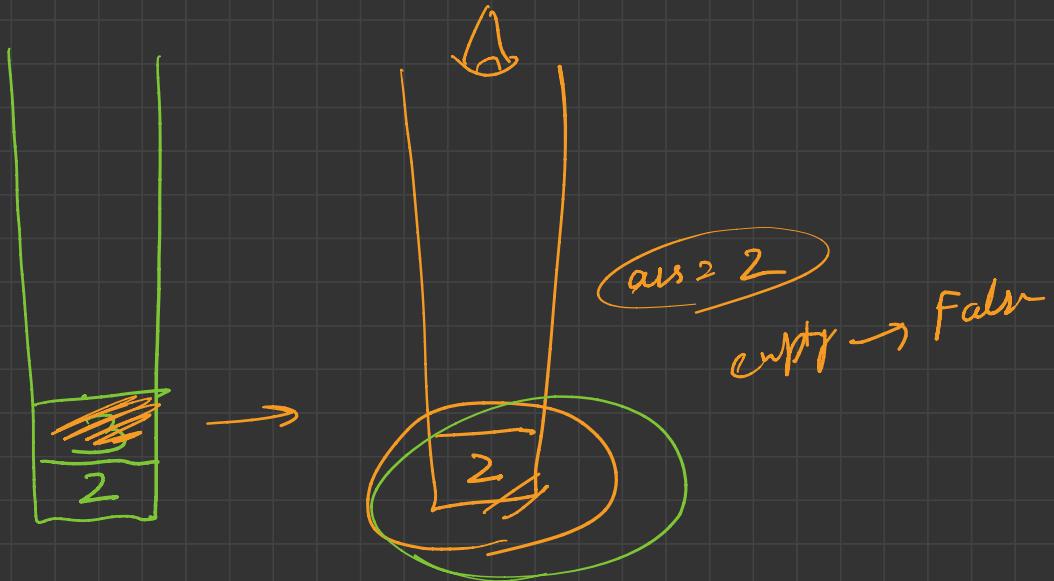
push \rightarrow

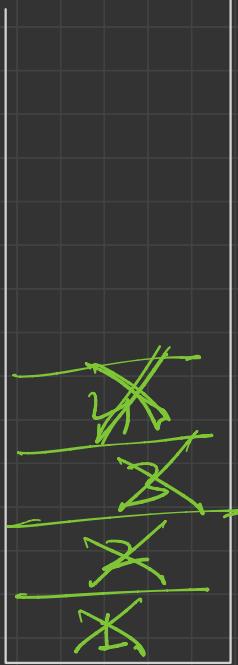
s.push (2);

pop \rightarrow

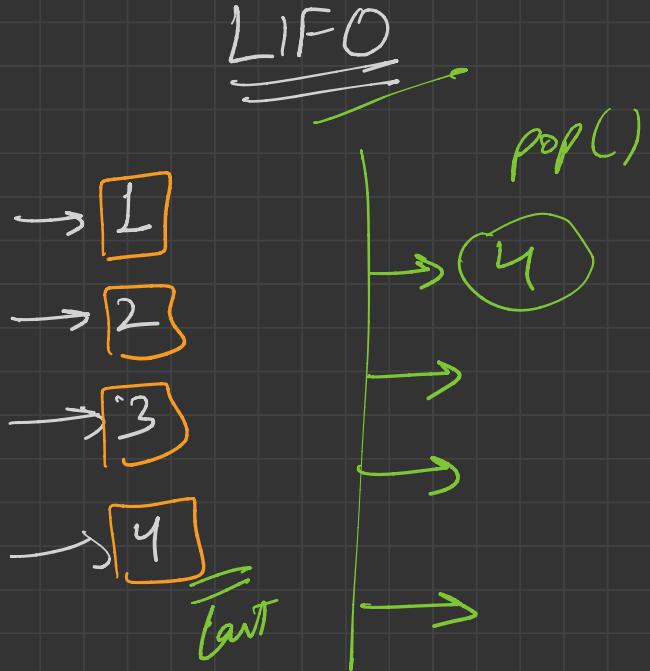
s.pop ();

Peek / top → s · top() → tip
is empty → s · empty() → T / F





Stack



→ what → ?

→ Stack we → code

STL

→ Stack Implementation

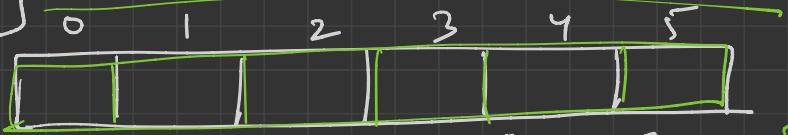
class stack

Stack → Impl

→ [array]

Linked List → Homework

top = -1



arr

push

int top → index

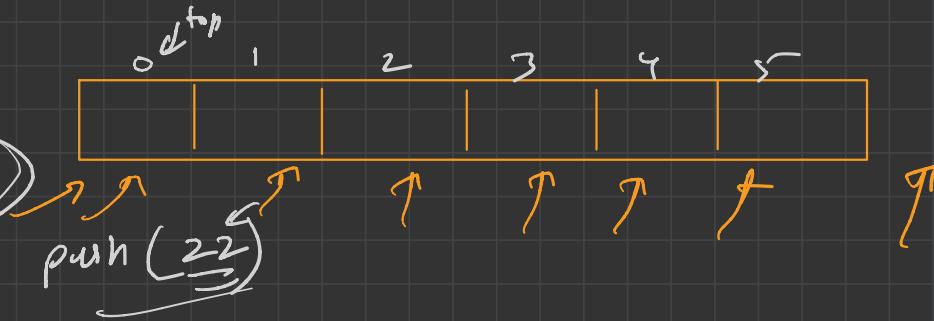
{
 int top = -1;
 int *arr = ...;
 int size = ...;

}

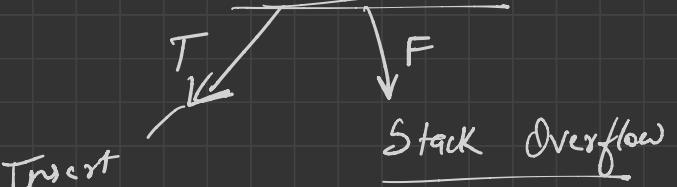
push operation

top = -1

top = 0



↳ check → is space available



Insert

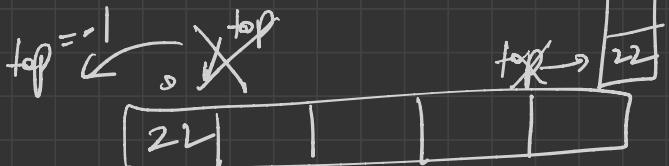
$\boxed{\text{top}++ ;}$
 $\boxed{\text{arr}[\text{top}] = 22}$

$\boxed{\text{top} = -1 \rightarrow}$ stack is empty

pop()

$\boxed{\text{top} \geq 0}$

check element is present



T ↘
 $\boxed{\text{top}--}$

→ pop()

Stack Underflow

`empty()` ↴

$\text{top} = -1 \rightarrow$ stack empty

`top()` ↴

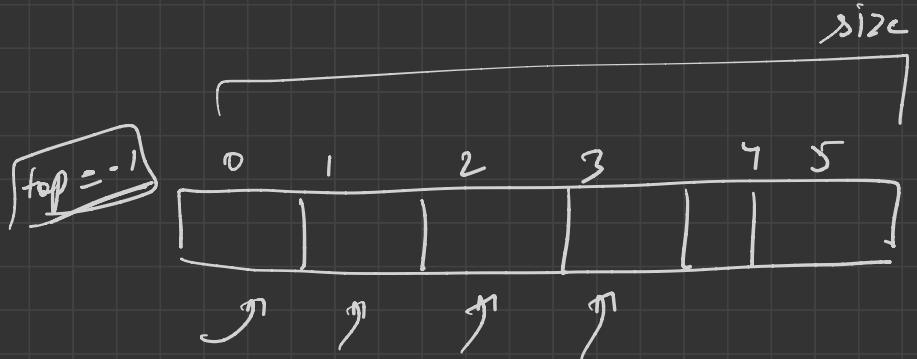
top ↴

$\boxed{0 \downarrow 2 2}$

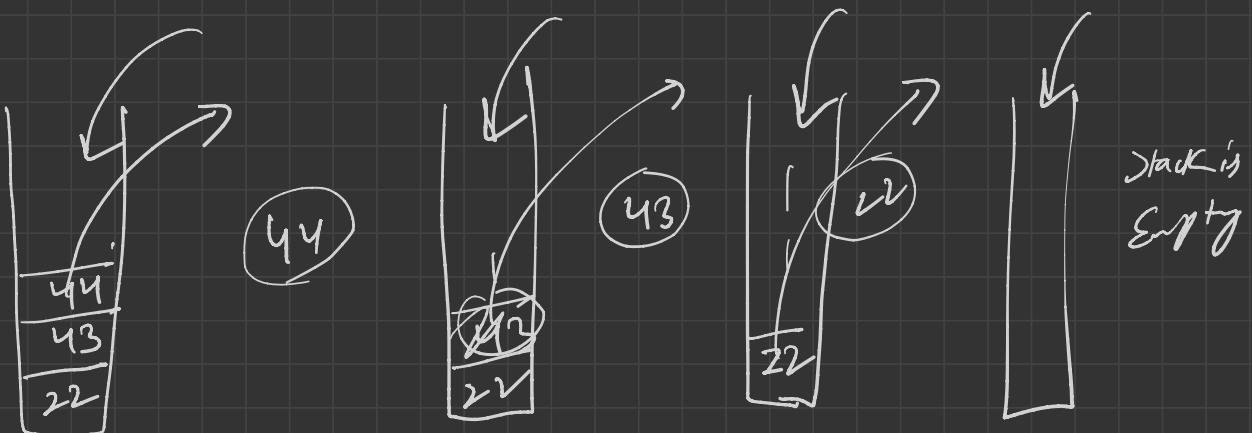
return $\text{arr}[\text{top}]$

$\text{top} = -1 \rightarrow$

$\text{top} < \text{size} \rightarrow$



$\xrightarrow{\quad}$ $\boxed{\text{size} - \text{top} > 1} \rightarrow \text{at least } \lceil \frac{\text{sept}}{2} \rceil \text{ place}$



→ Stack top → array

LL → Homework

Easy

→ Easy Question

↳ 2 stacks in an array

→ Approach

$top1 = -1$

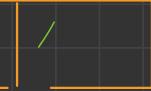
OR



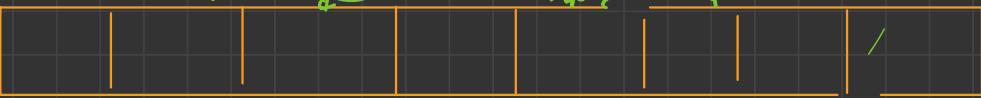
Single arry

$top1 \rightarrow$

$top2 \rightarrow$



$top2 = size$



1 way

$\xrightarrow{\text{stack } 1}$

$n/2$

$n/2$

$\xrightarrow{\text{stack } 2}$

$n/2$

$\xrightarrow{\text{spac}}$

$n/2$

→ Space not
optimally utilized

