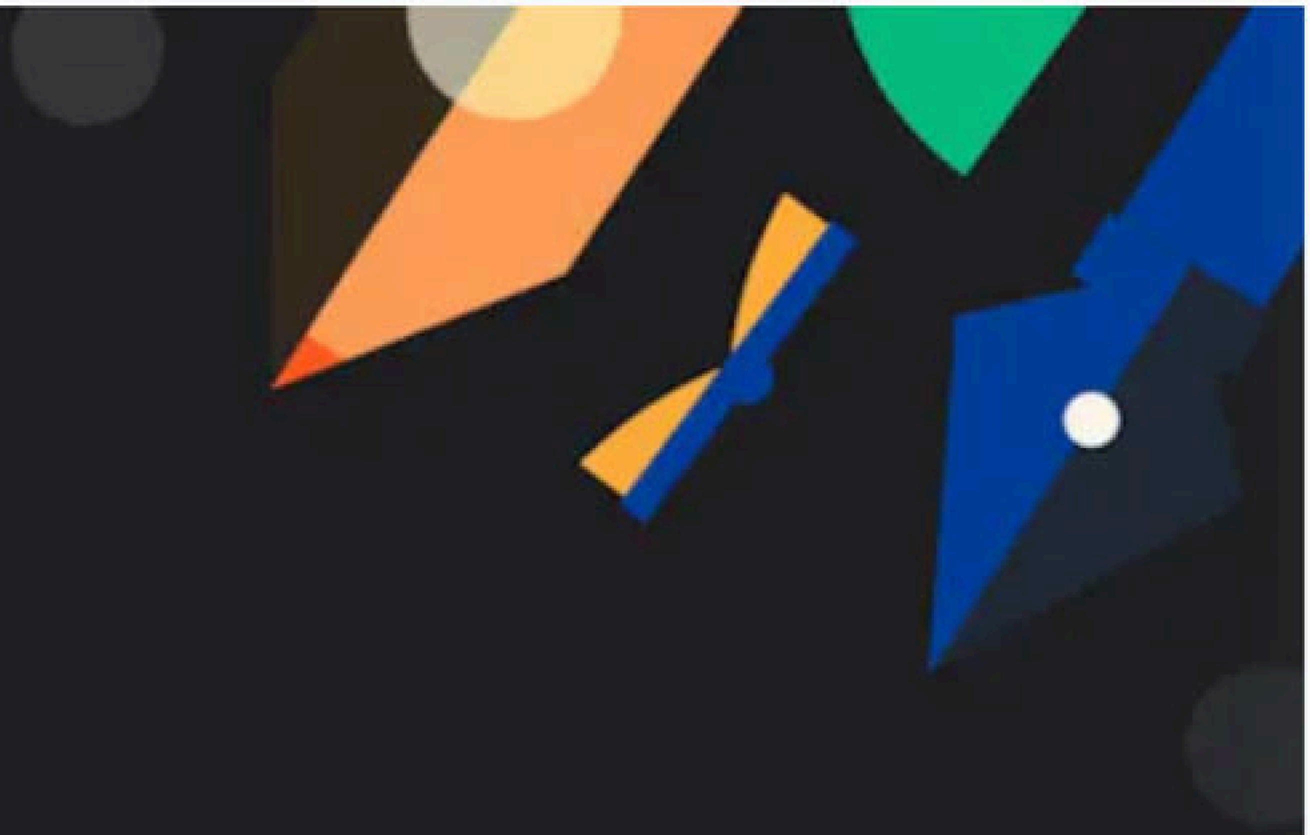


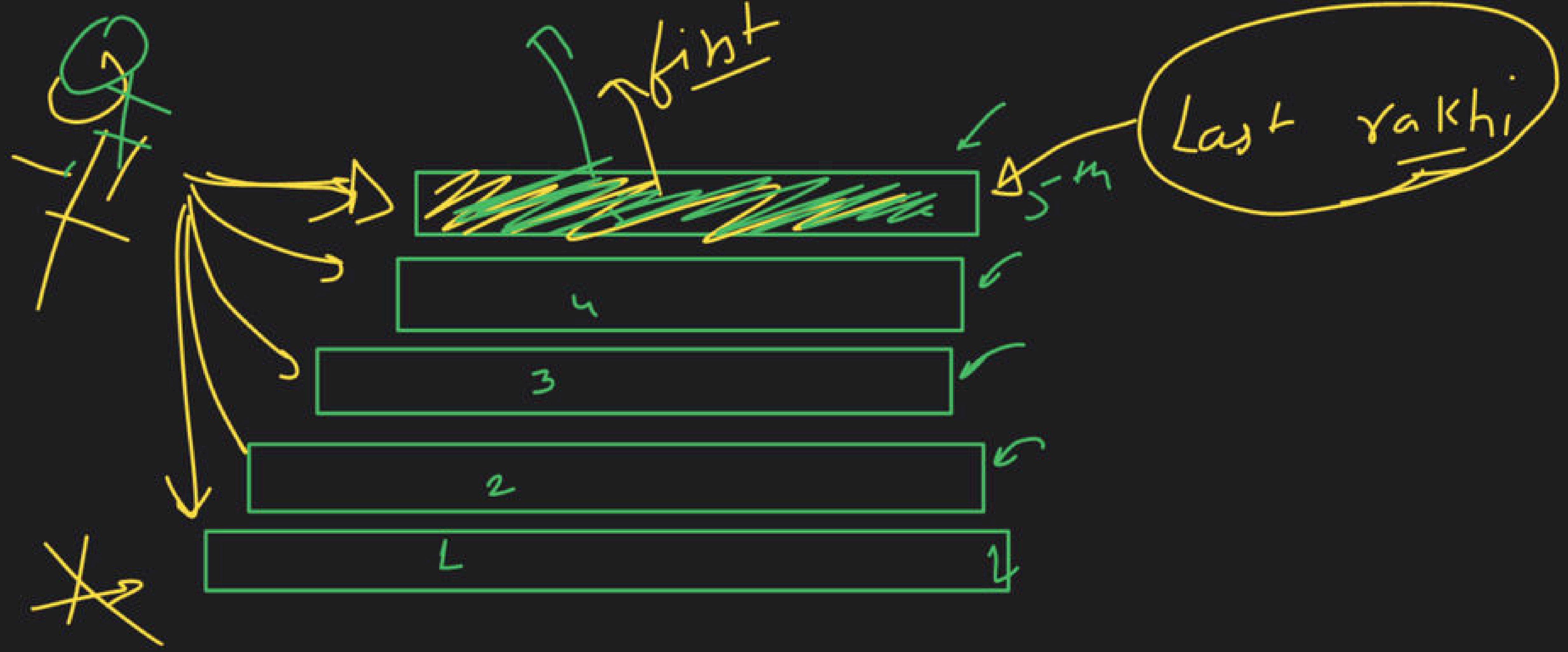
Stack Class - 1

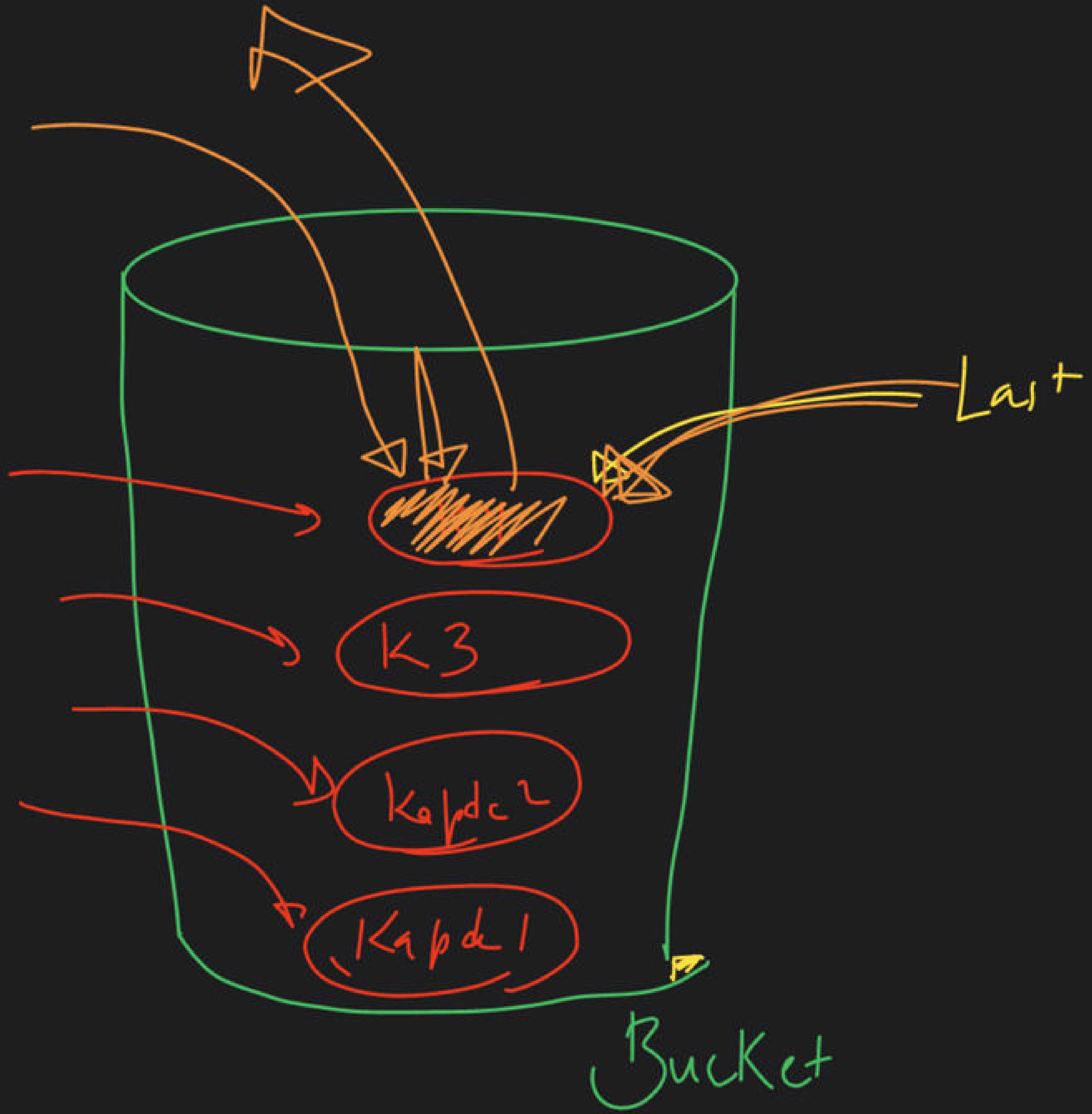
Special class

Love Babbar • Nov 8, 2023

→ Stack: → Data Structure → LIFO

(Last in first Out)




Last
K3
Kapdc2
Kapdc1
Bucket

Stack Empty

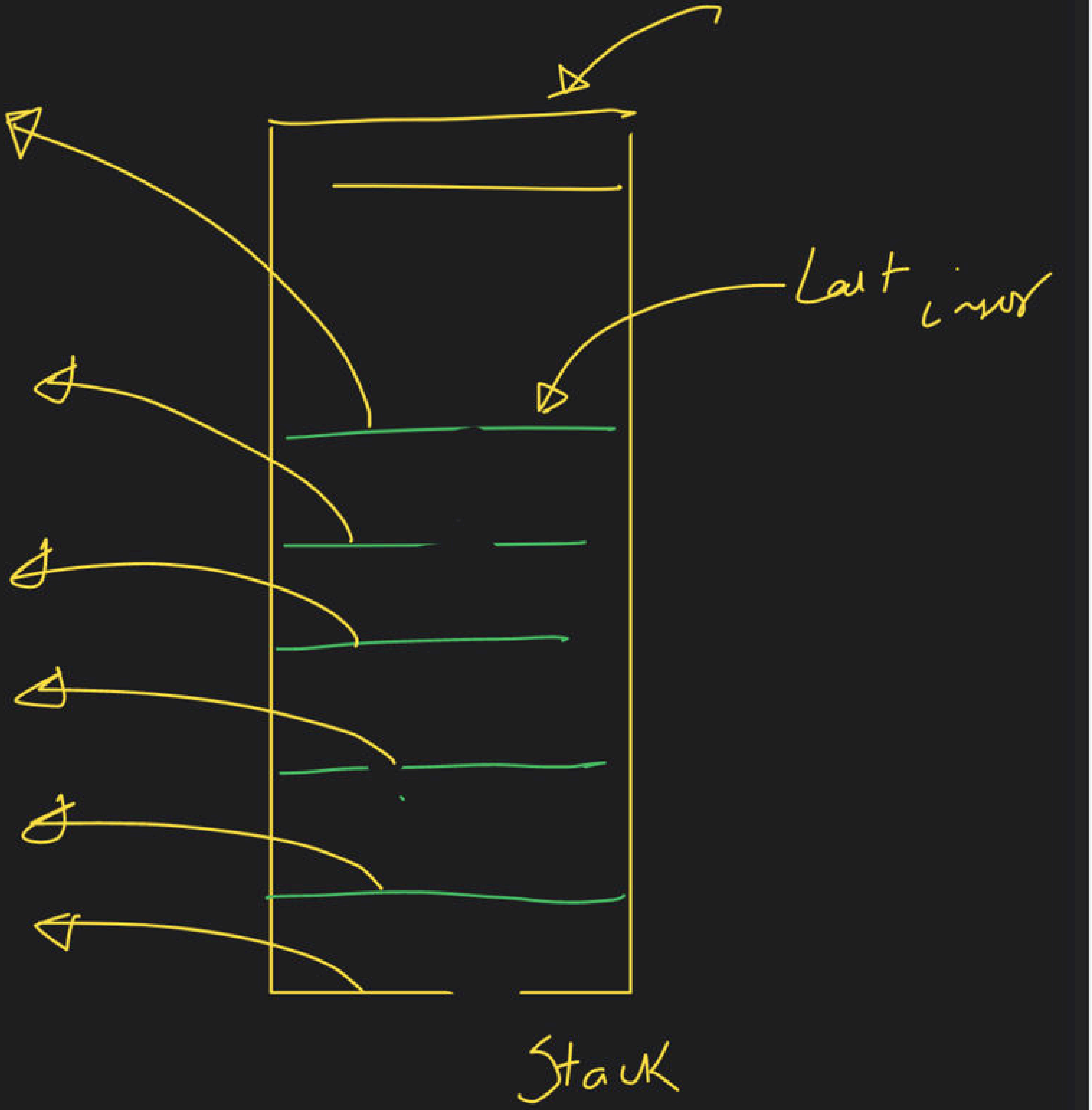
Remove

Underflow

Stack - full

Insert

Overflow



MS word

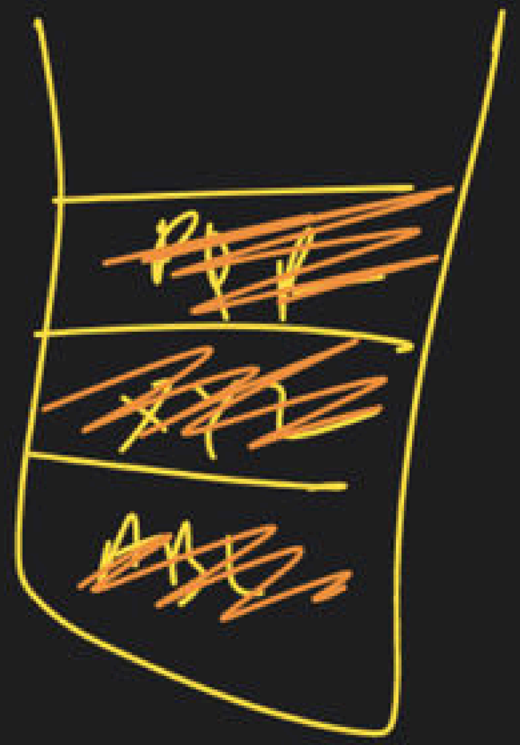
→ Abcd

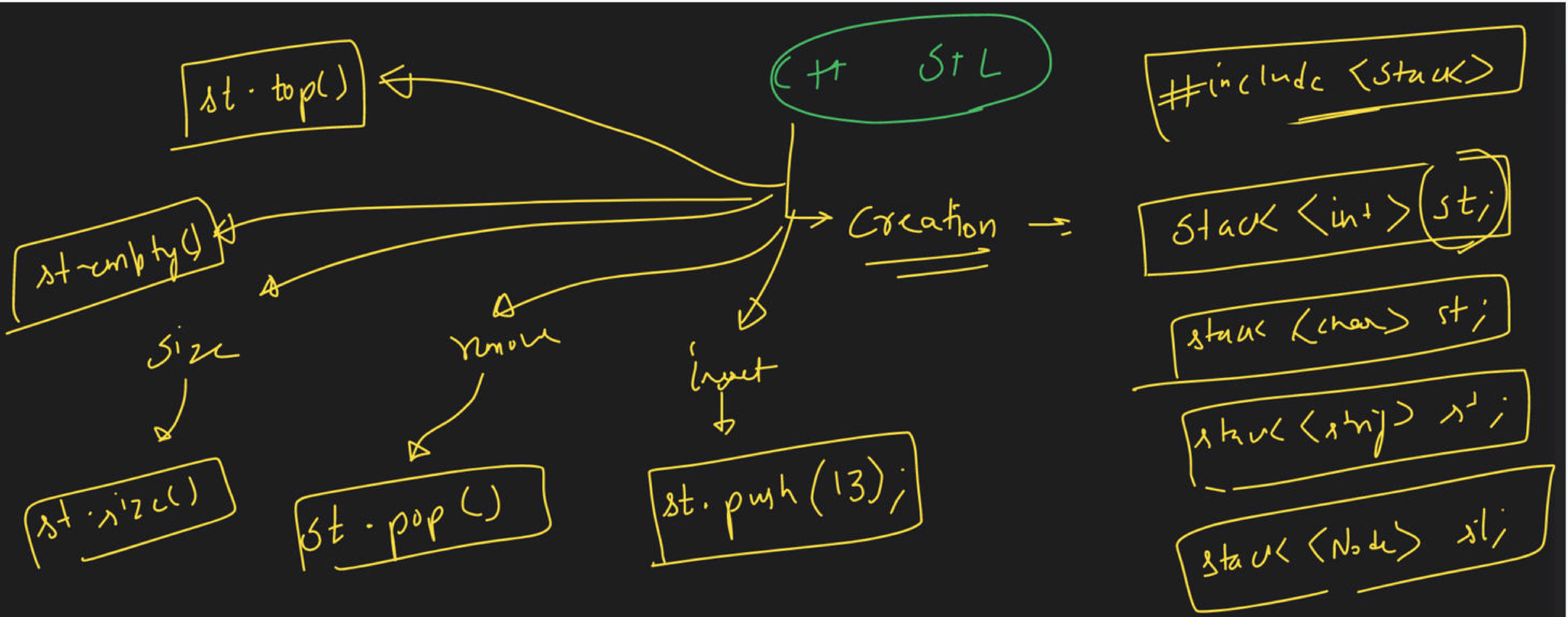
→ XYZ

→ PQR

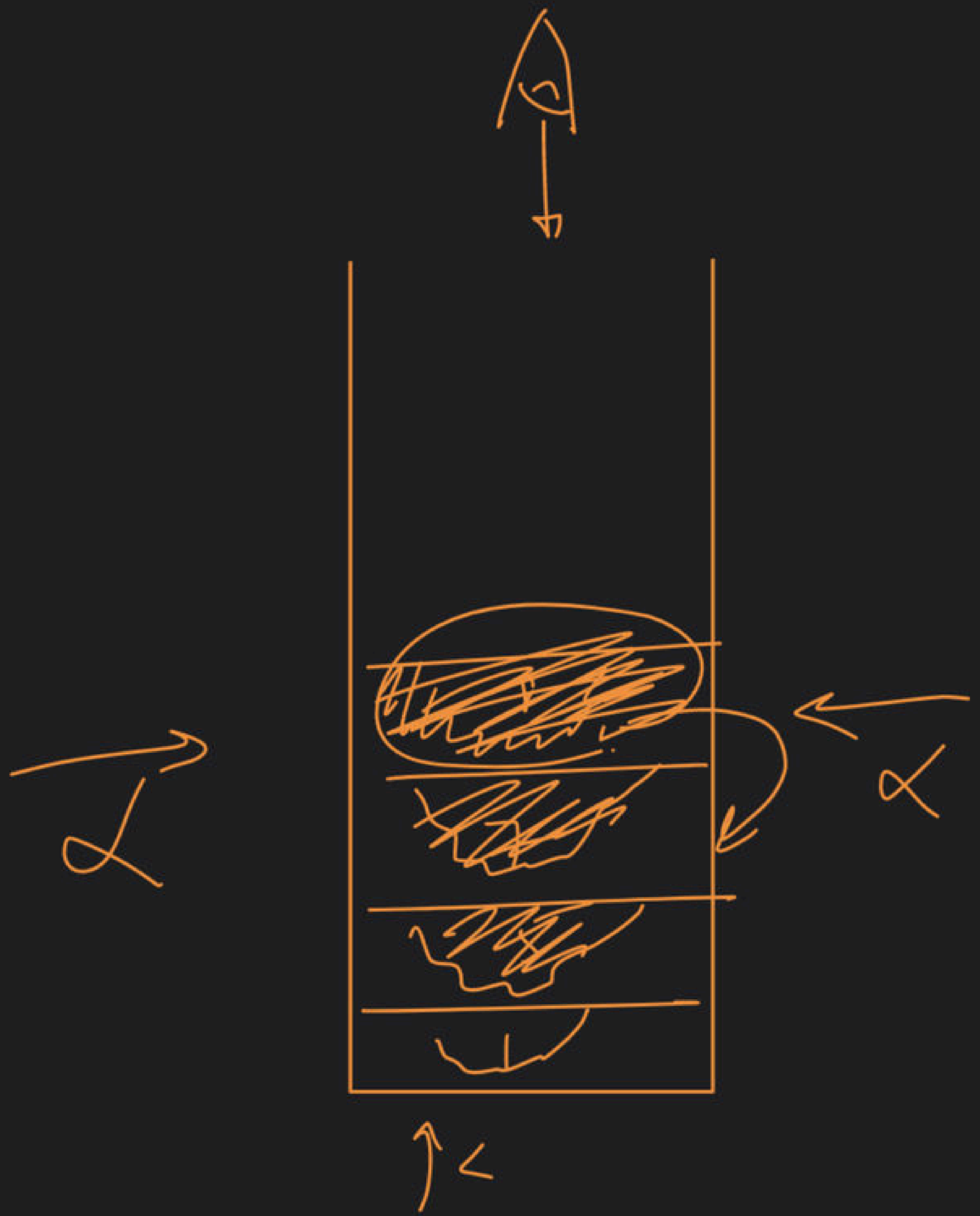
Ctrl + Z

→ Undo

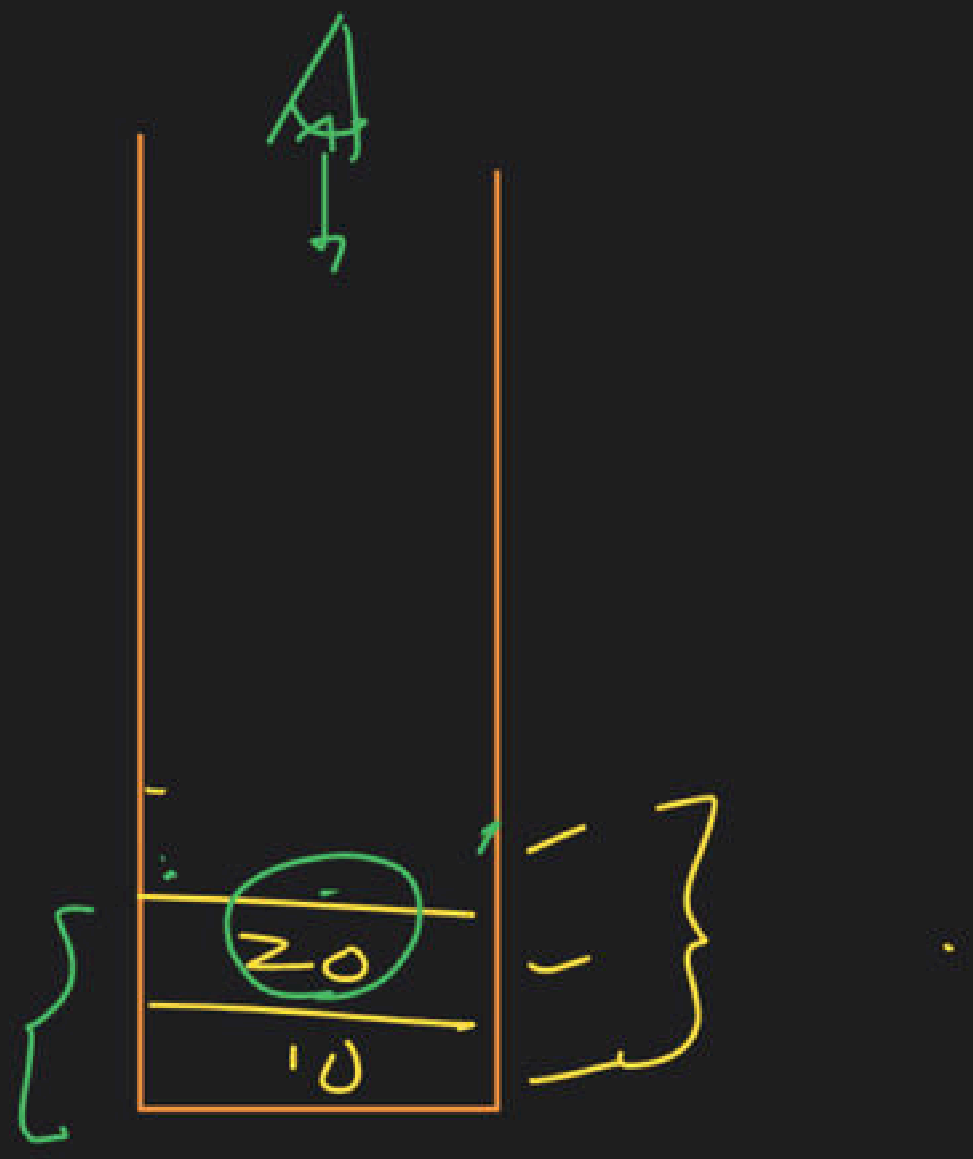




```
#include <stack>

stack<int> st;
stack<char> st;
stack<string> st;
stack<Node> st;

st.push(13);
st.pop()
st.size()
st.empty()
st.top()
```



```
stack <int> st;

st.push(10);
st.push(20);
st.push(30)

cout << st.size(); → 2

st.pop()

if (st.empty())
    cout << Empty;
else
    cout << Not Empty;

cout << st.top(); → 20
```

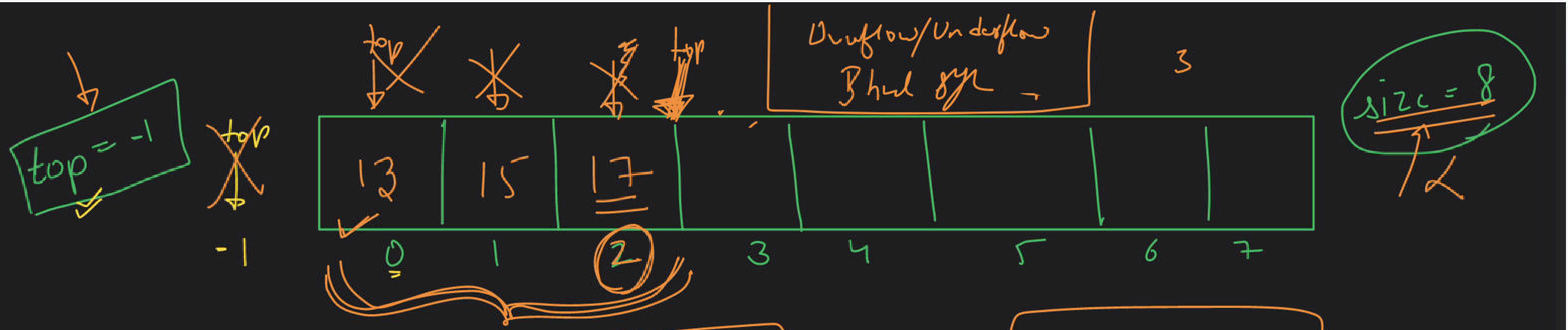


Stack → Implementation

Stack →

- getSize()
- check empty()
- push()
- pop()
- get Top()

Stack → dynamic array



push(13)
→ top++;
→ arr[top] = data

pop()
→ top--

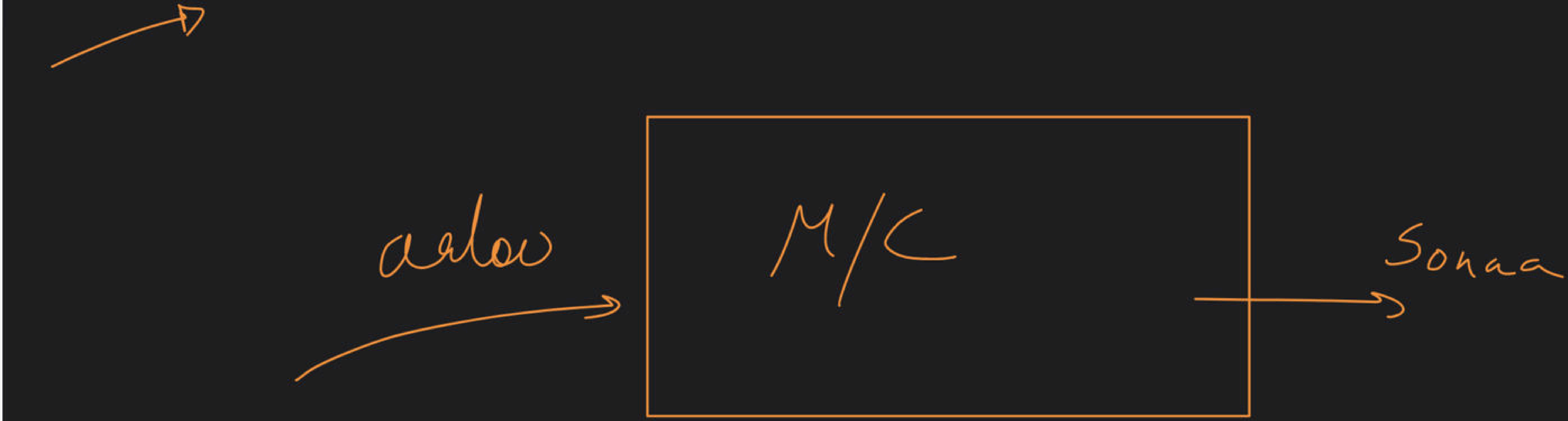
isEmpty()
→ top == -1
→ Empty

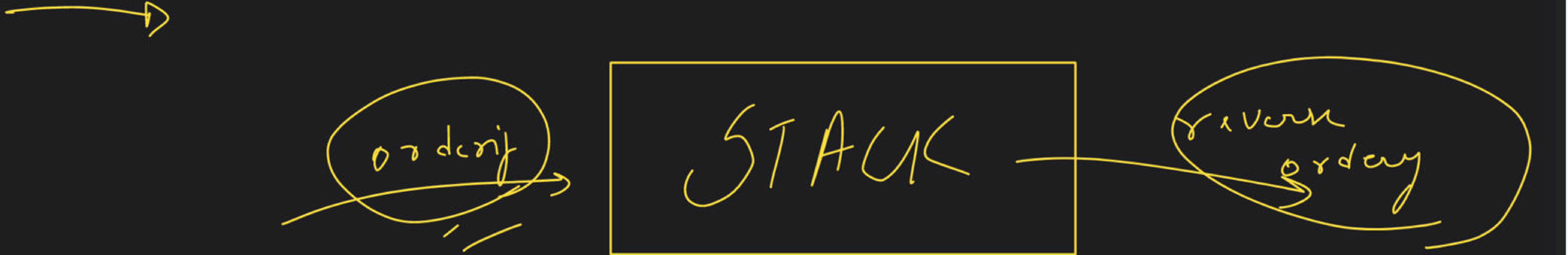
getTop()
→ return arr[top]

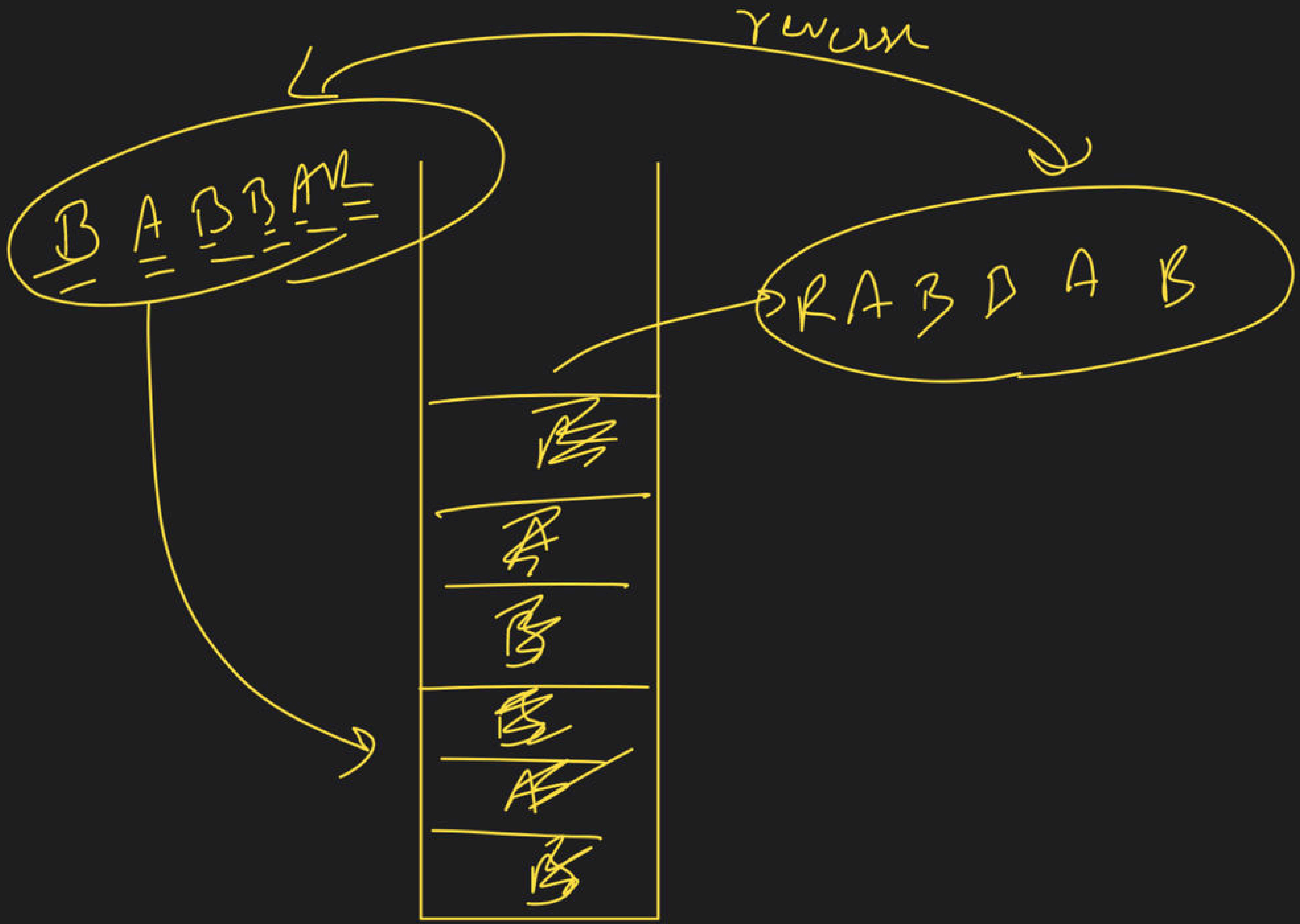
getSize() → no of elmt in stack
→ return top+1

```
class Stack
{
        int *arr;
        int size;
        int top;

      Stack()

      getTop()
        push()
        pop()

      isEmpty()
      getSize()


};
```

→ Diwali

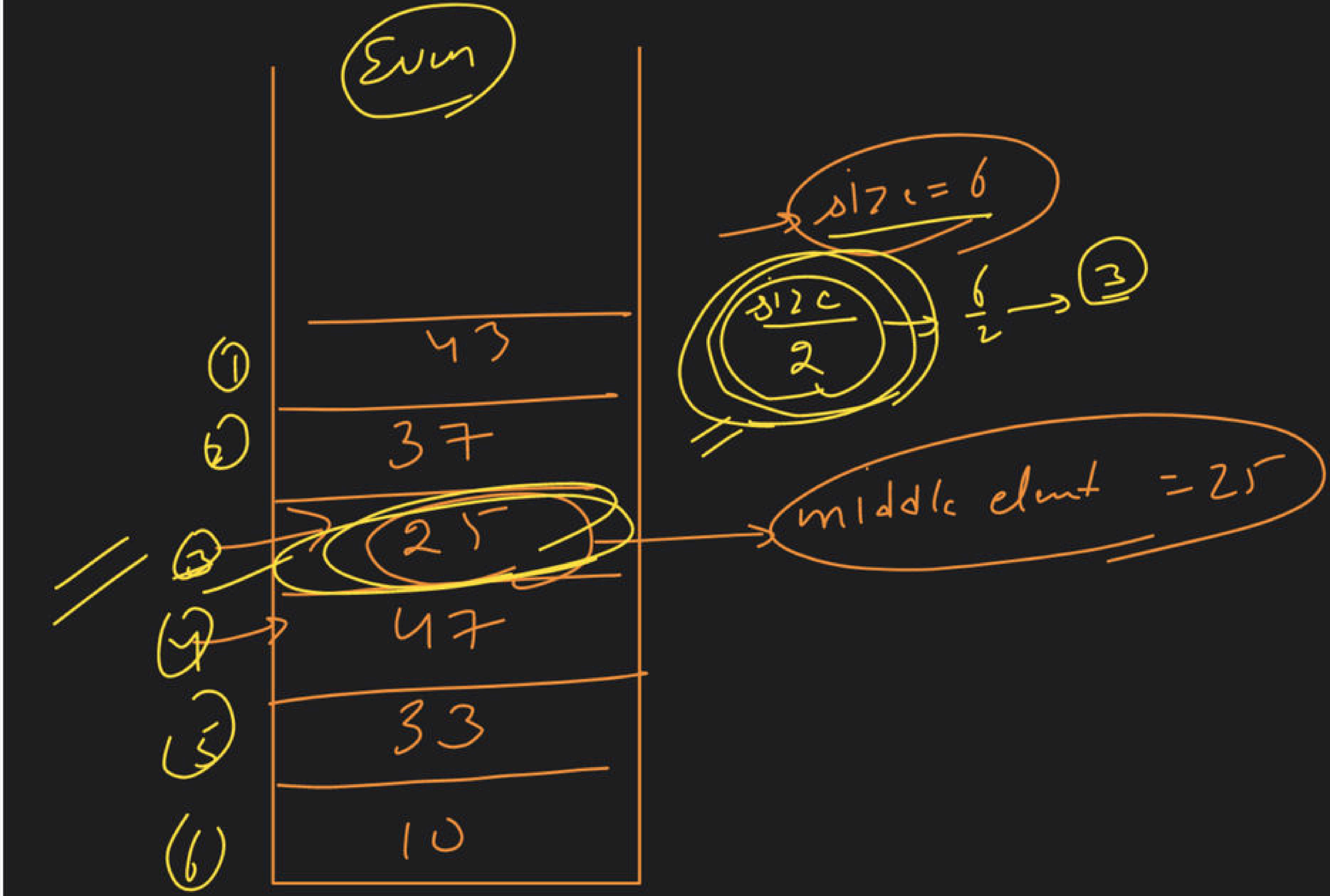

aalu
M/C
Sonaa

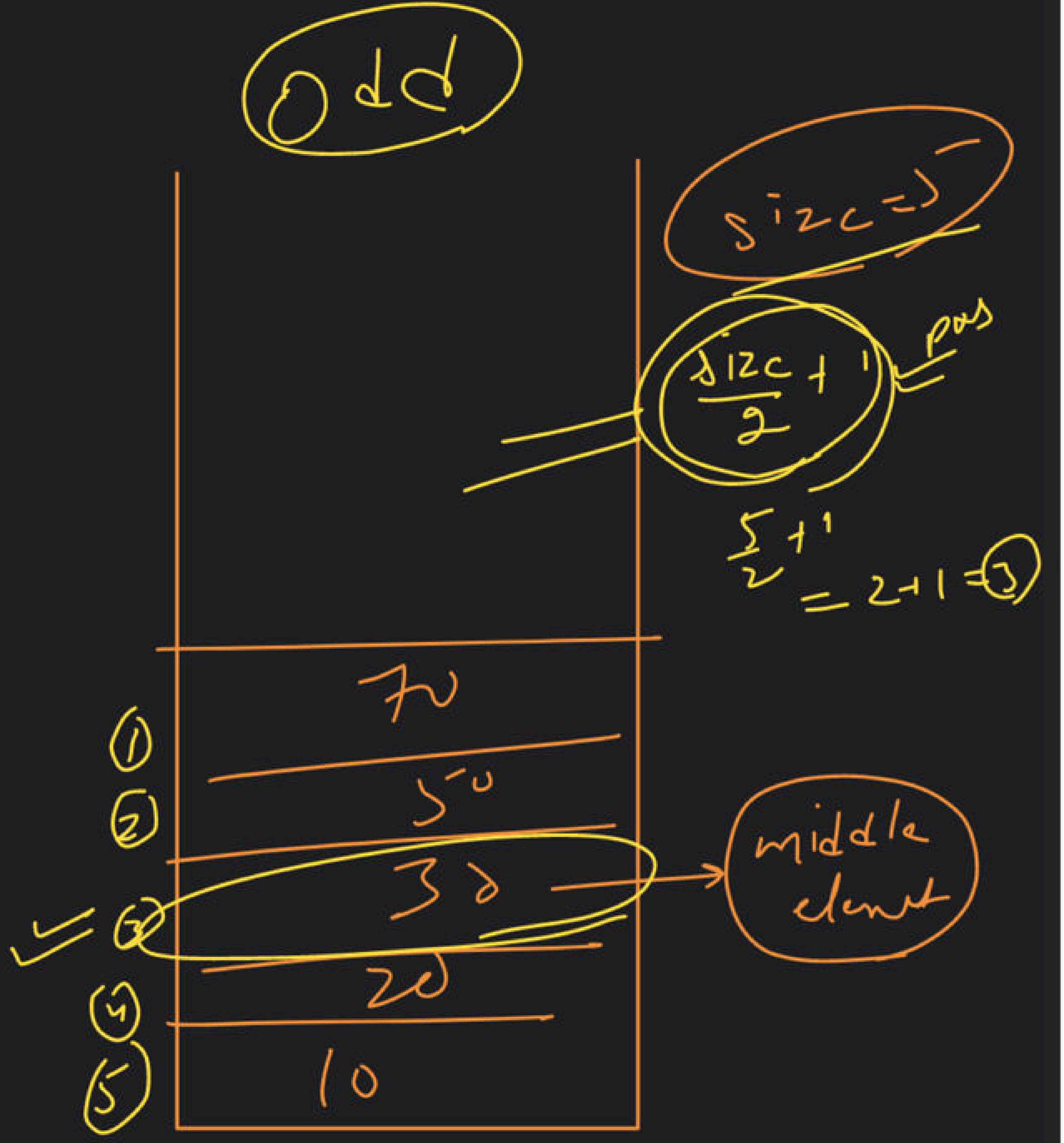

ordering
STACK
reverse order

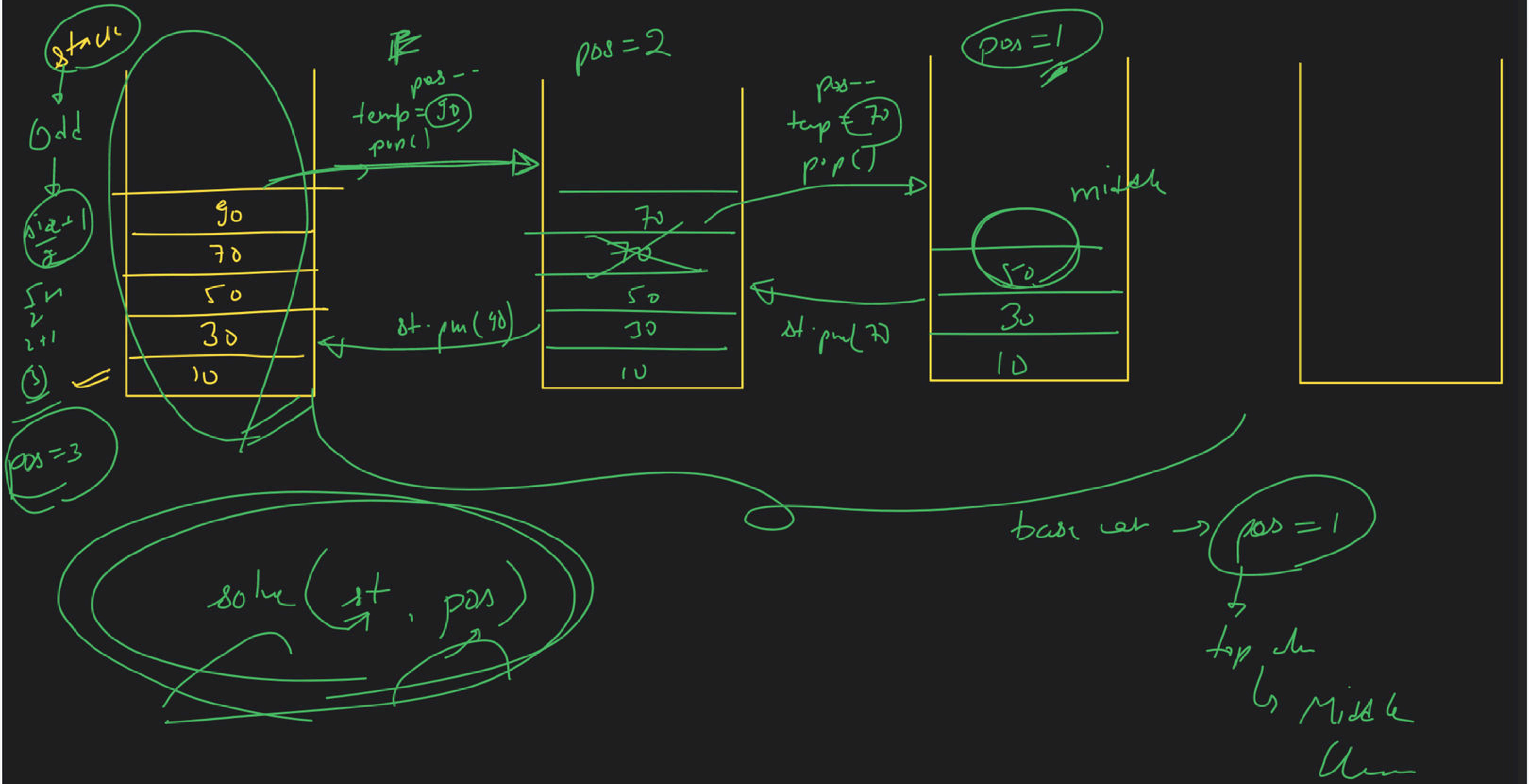


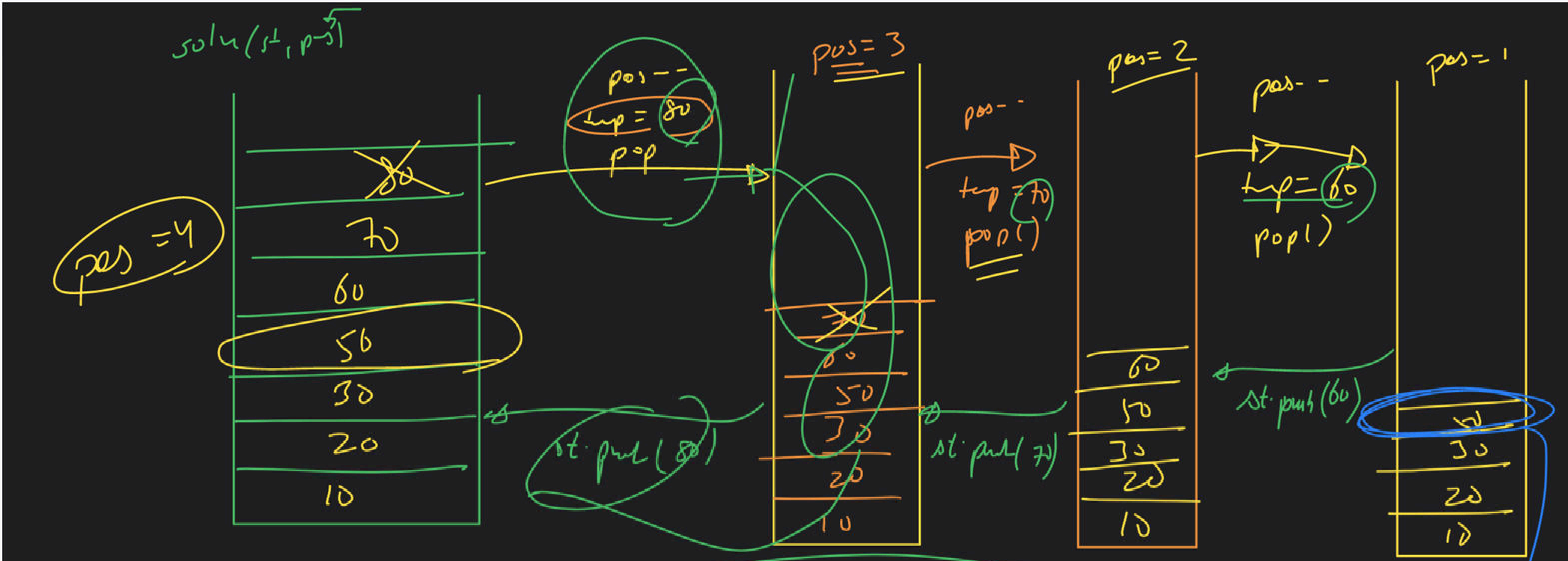
String
↳ "BABBAR"

Middle Element of a Stack :-)






stack
odd
size+1 / 2
5+1 / 2
2+1
3
pos=3
90
70
50
30
10
pos--
temp = 90
pop()
st.push(90)
pos=2
70
70
50
30
10
pos--
temp = 70
pop()
st.push(70)
pos=1
middle
50
30
10
base case → pos=1
→ top ele
↳ Middle elem
solve(st, pos)



Even → pos = $\frac{size}{2}$

Odd → pos = $\frac{size}{2}+1$

$\frac{7}{2}+1 = 4$

B.C → pos == 1 → ans found

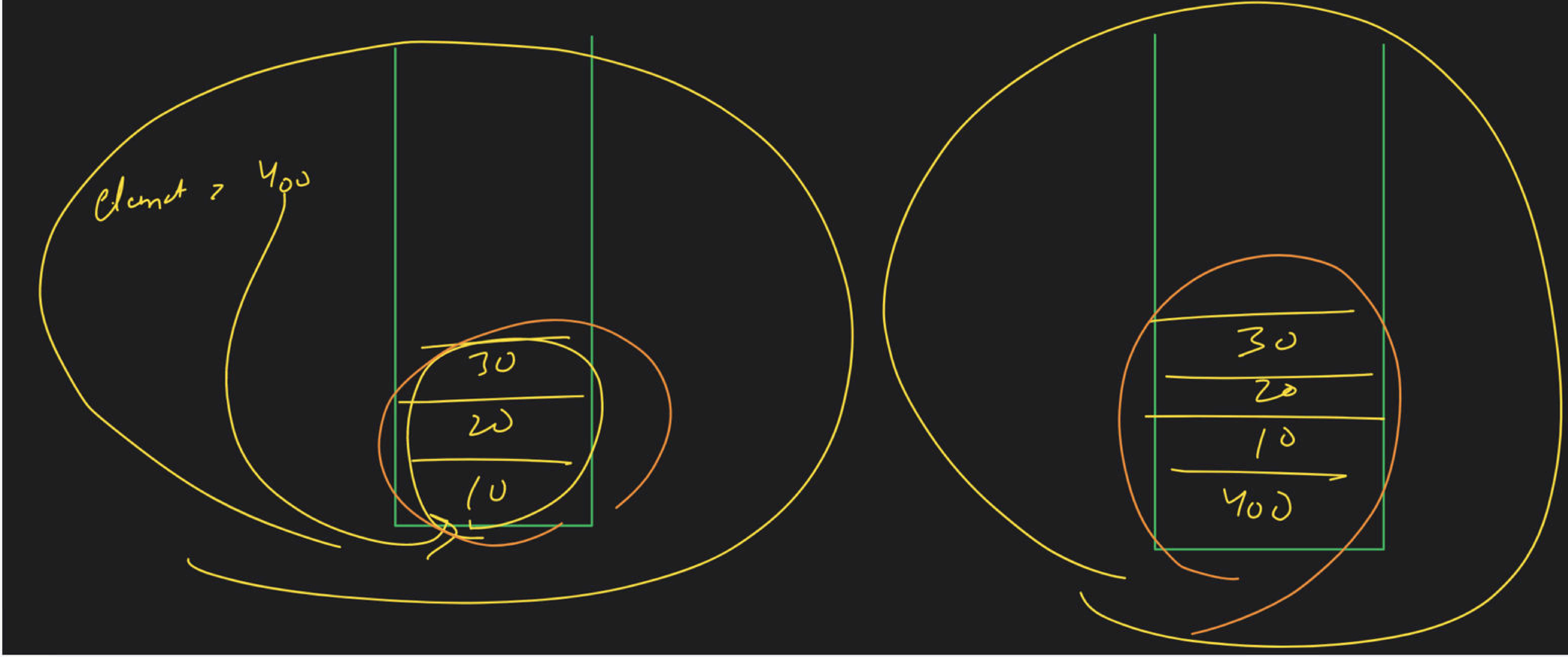
pos--

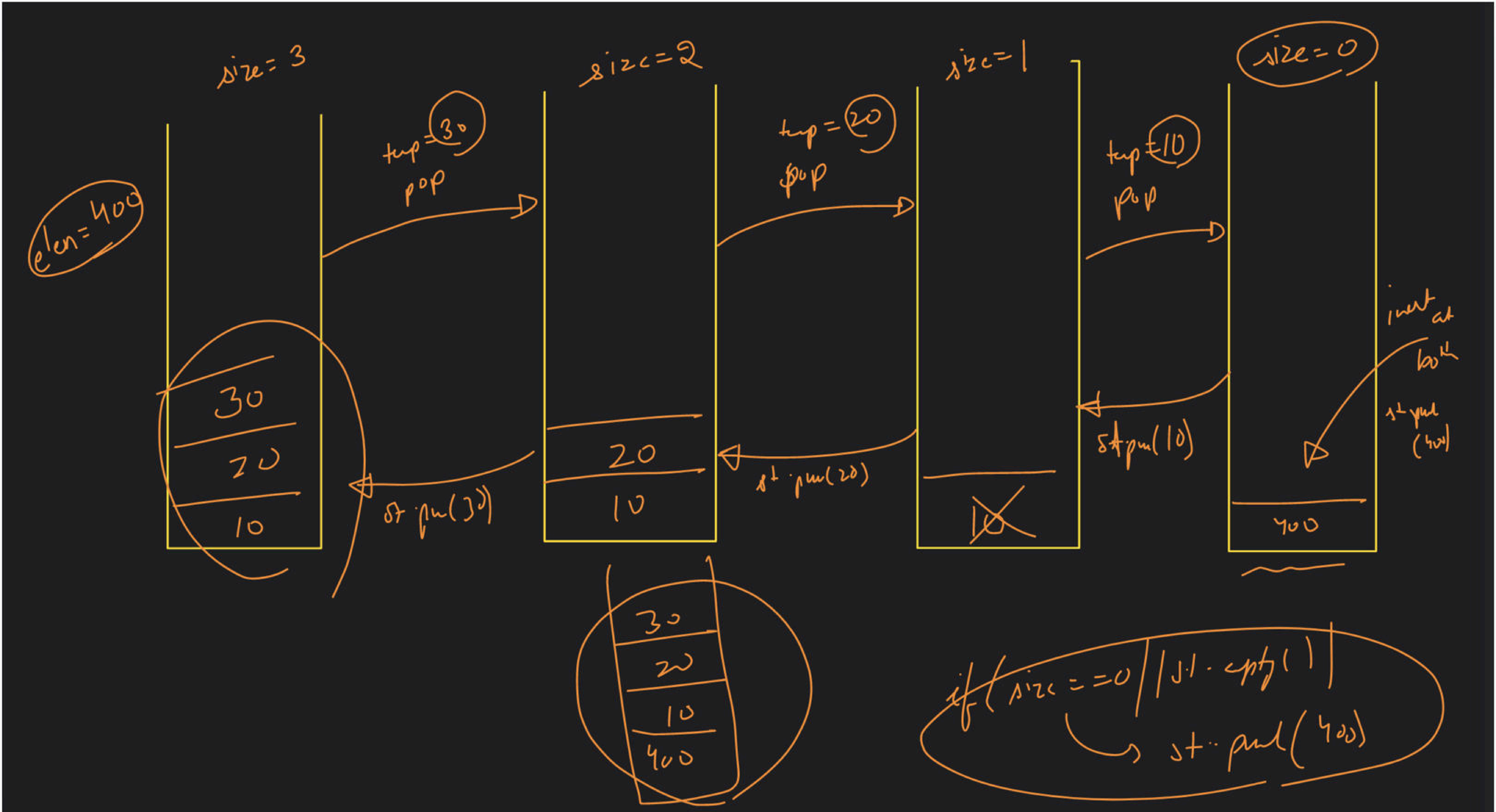
temp = st.top()

st.pop() → Rec → st.push(temp)

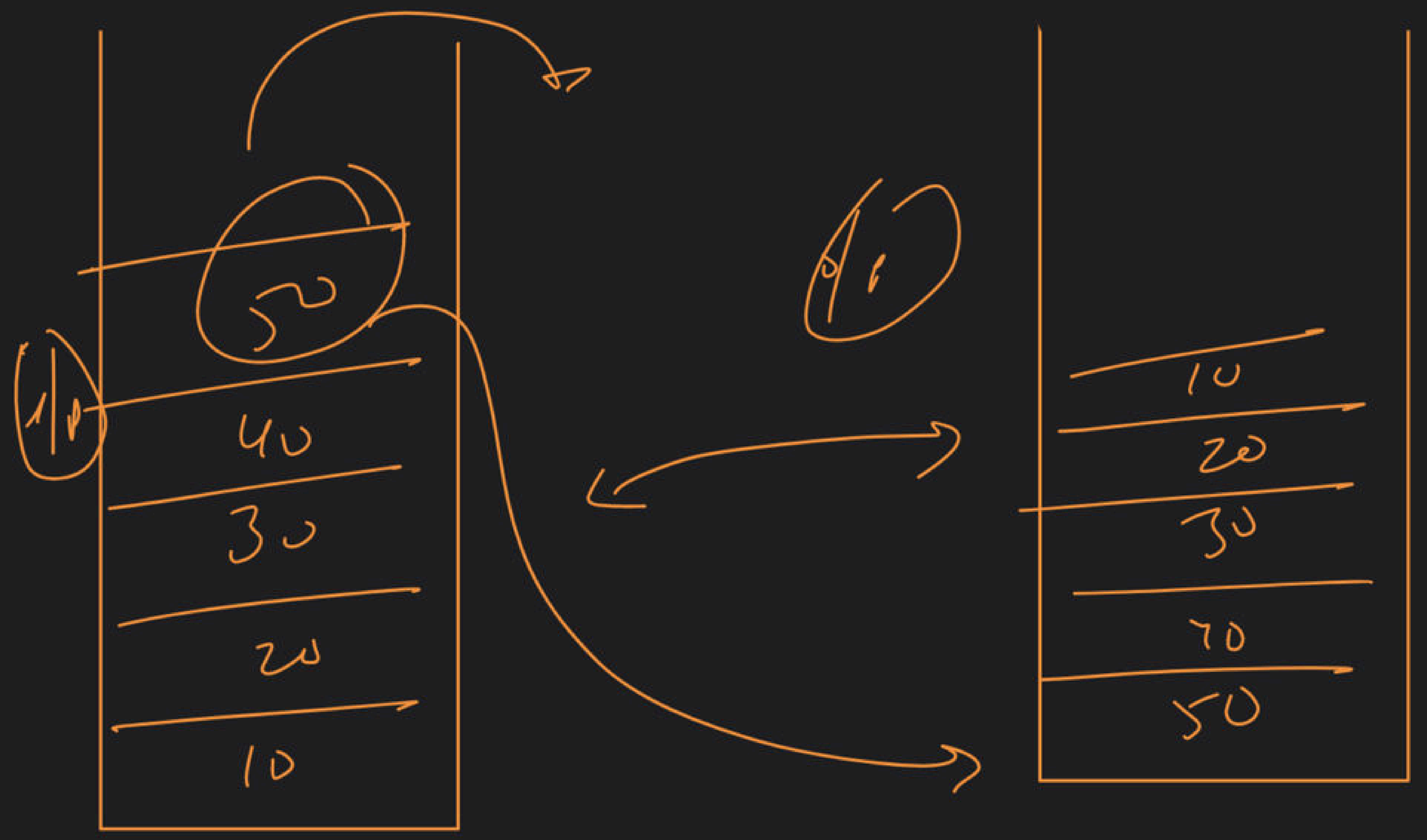
ans=50

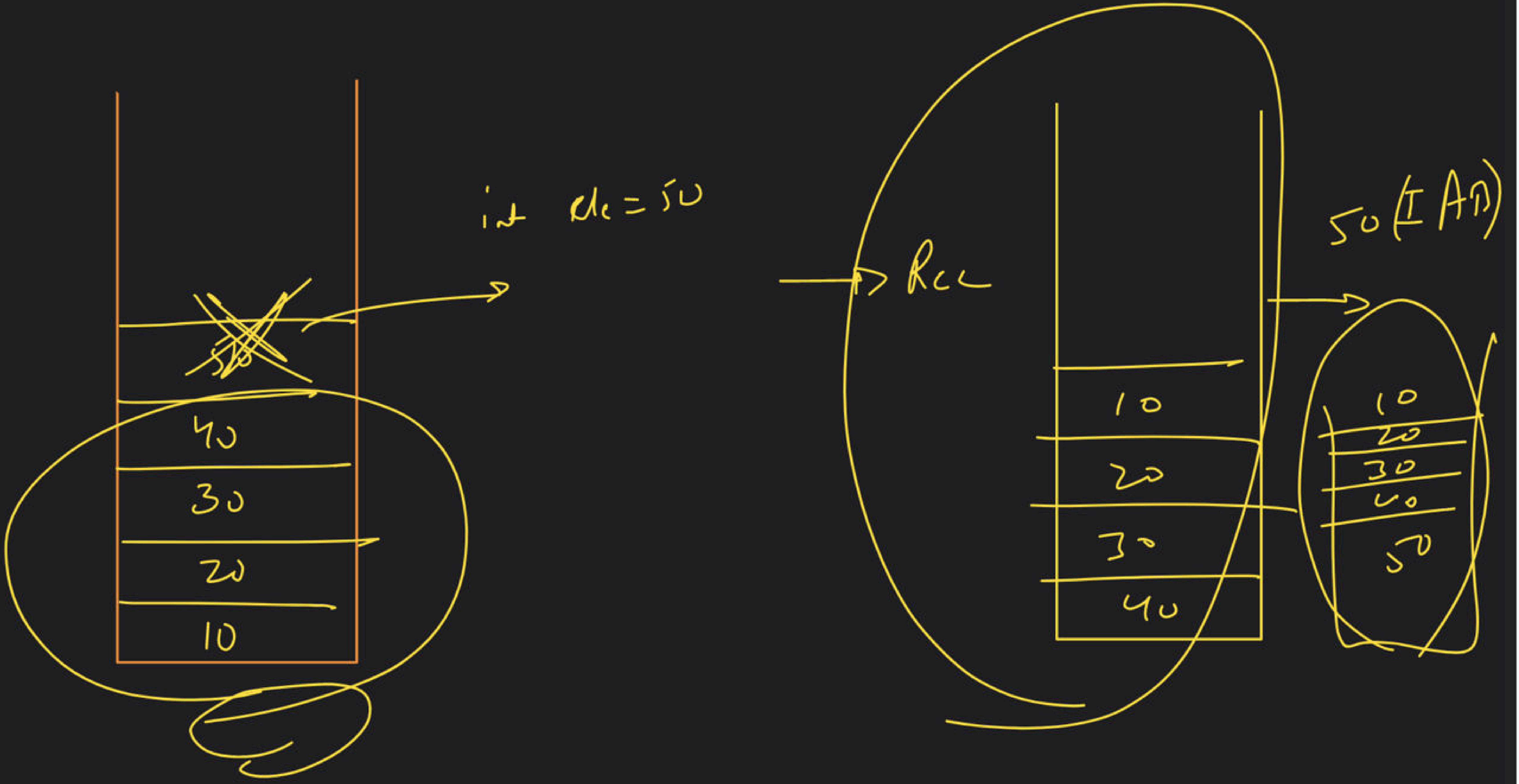
Insert at bottom of Stack



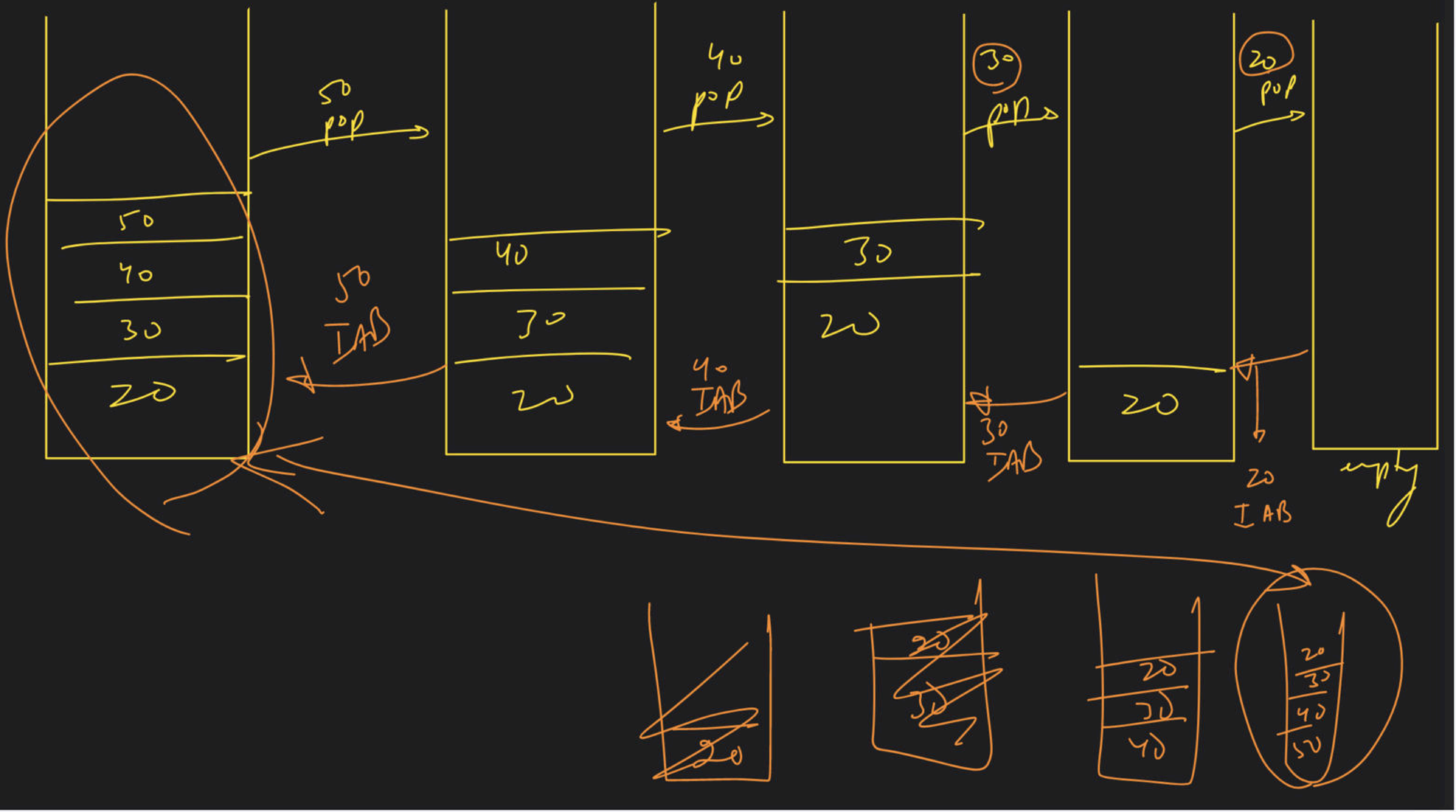

size = 3
elen = 400
30
20
10
top = 30
pop
st.push(30)
size = 2
20
10
top = 20
pop
st.push(20)
size = 1
10
top = 10
pop
st.push(10)
size = 0
insert at back
st.push (400)
400
30
20
10
400
if (size = = 0 || st.empty ( ))
st.push (400)

→ Reverse a Stack #1



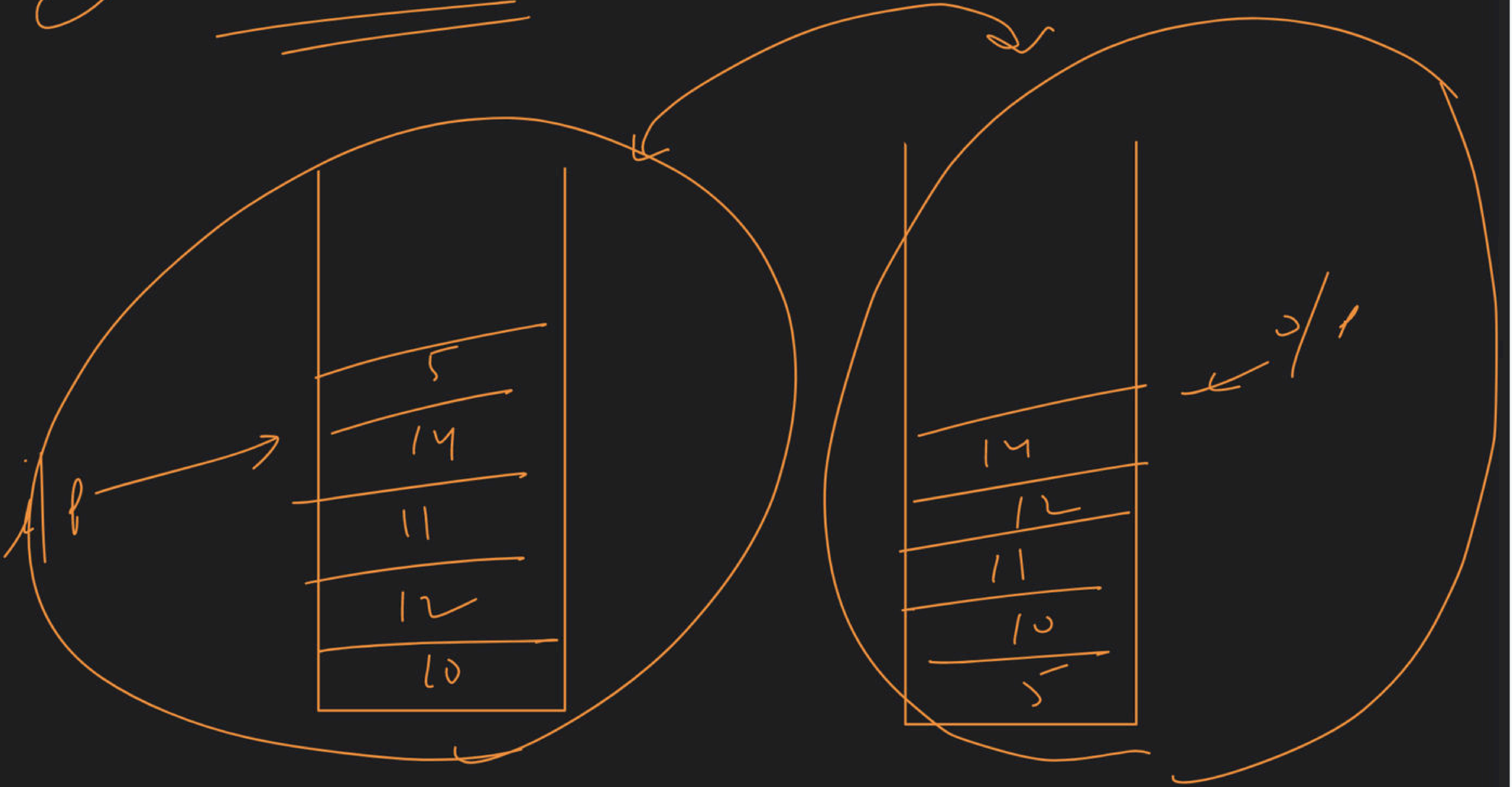


~~50~~
40
30
20
10
int ele = 50
Rec
10
20
30
40
50 (IAT)
10
20
30
40
50

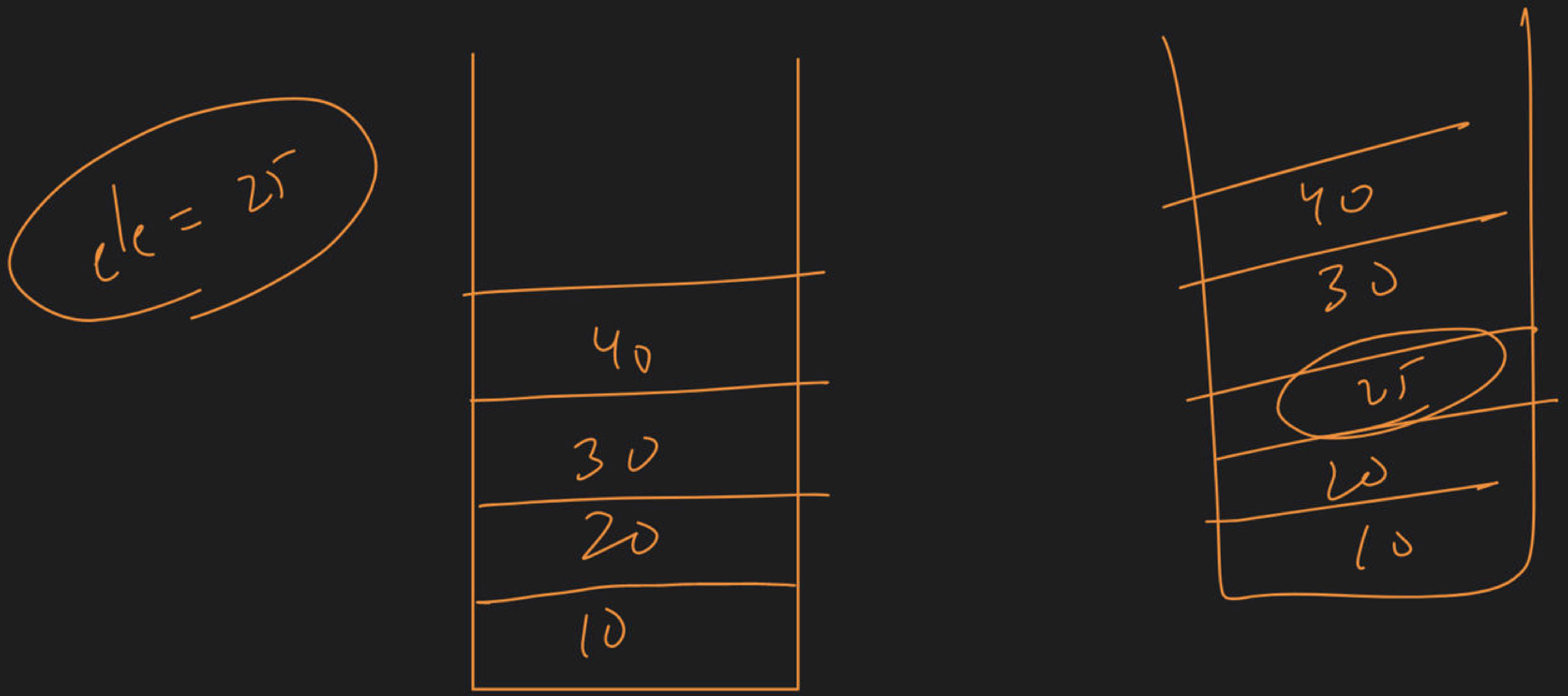

50
40
30
20
50
pop
40
30
20
50
IAB
40
pop
30
20
40
IAB
30
pop
20
30
IAB
20
pop
20
IAB
empty
20
20
30
20
30
40
20
30
40
50

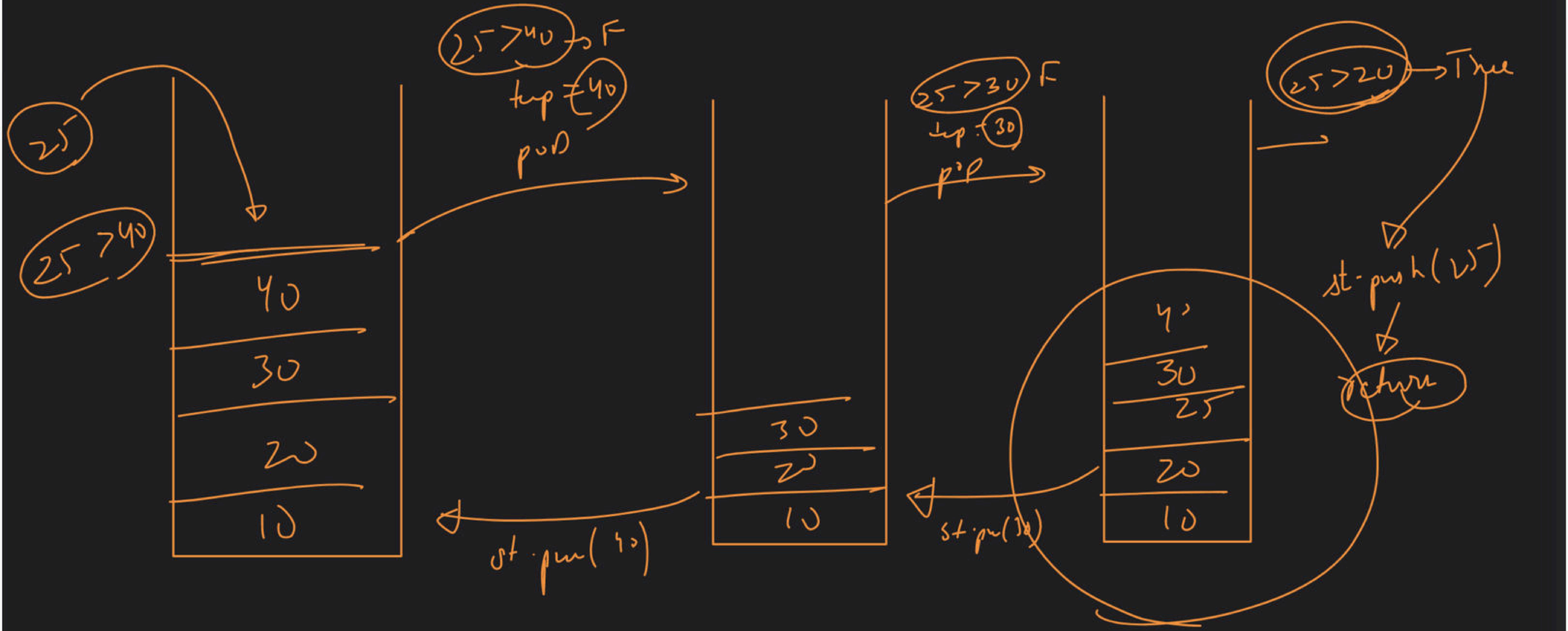
→ Sort a Stack

How 2 min



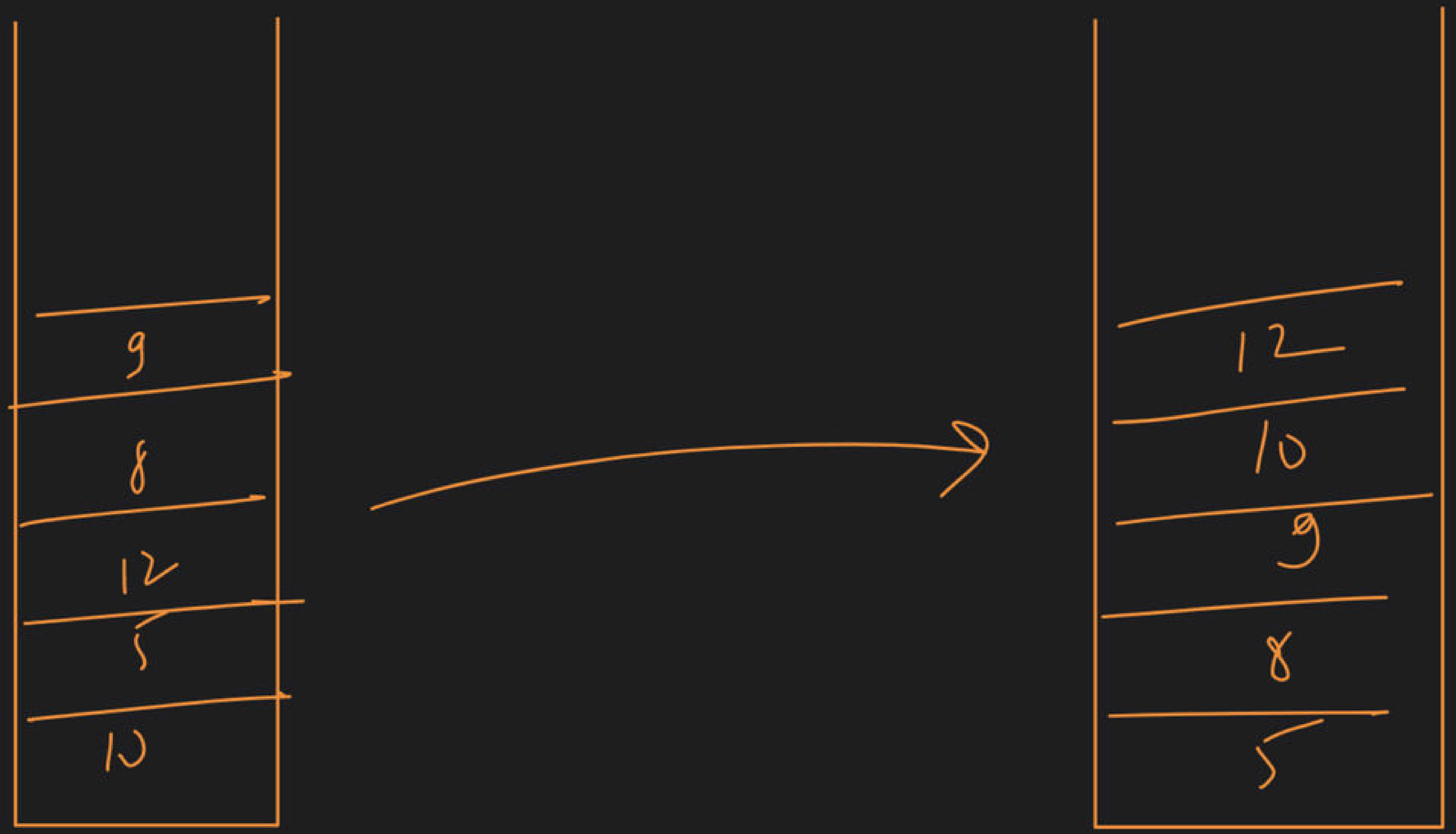
insert in a Sorted ~~Array~~ Stack

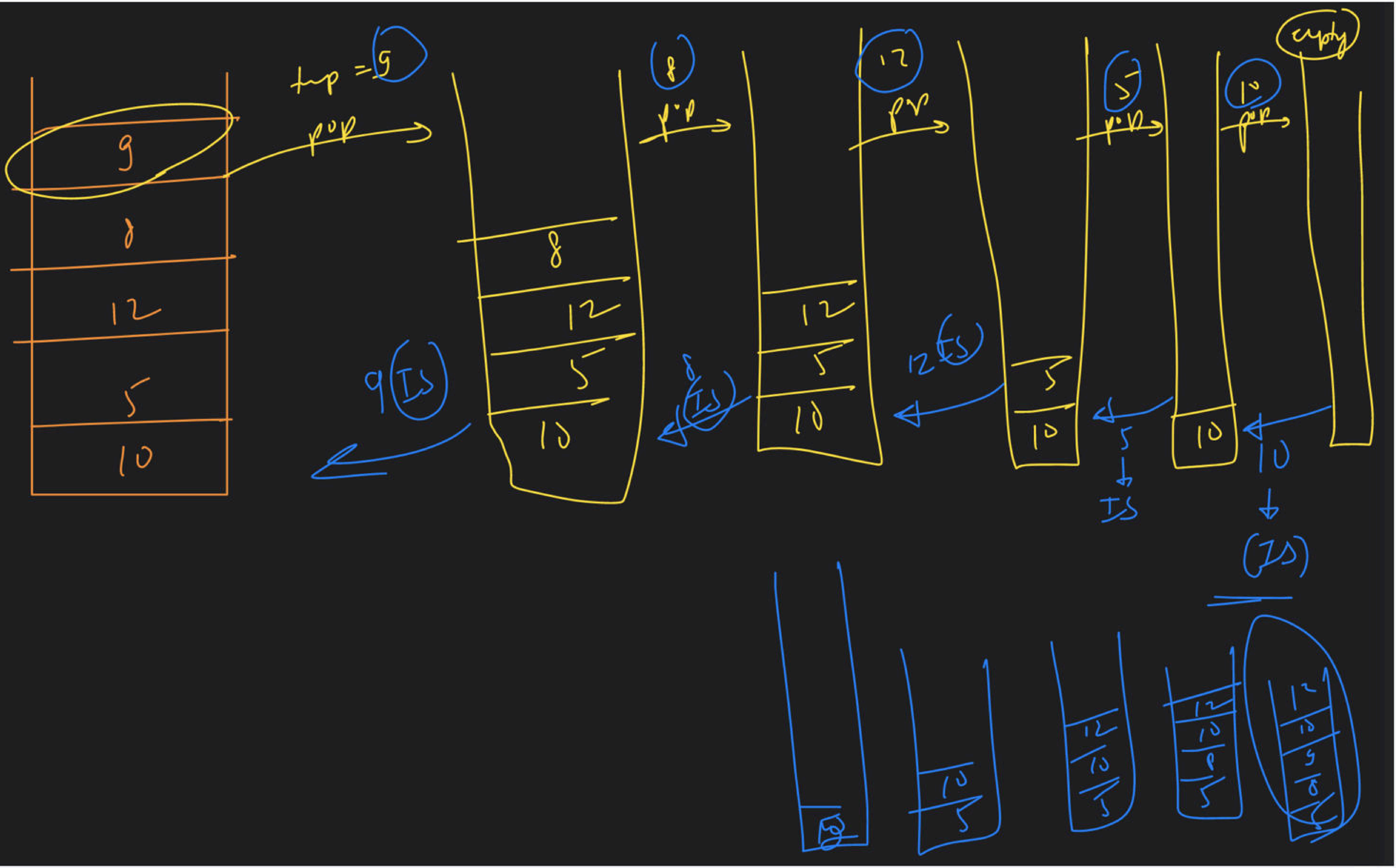



25
25 > 40
40
30
20
10
25 > 40 → F
top = 40
pop
st.push(40)
25 > 30 F
top = 30
pop
30
20
10
st.push(30)
25 > 20 → True
st.push(25)
return
40
30
25
20
10

//TC

Soot a Stack:-




9
8
12
5
10
tmp = 9
pop
9 IS
8
12
5
10
8
pop
8 IS
12
5
10
12
pop
12 IS
5
10
5
pop
5
IS
10
10
pop
10
(IS)
empty
9
10
5
12
10
5
12
10
8
5
12
10
9
8
5