



Stack - I

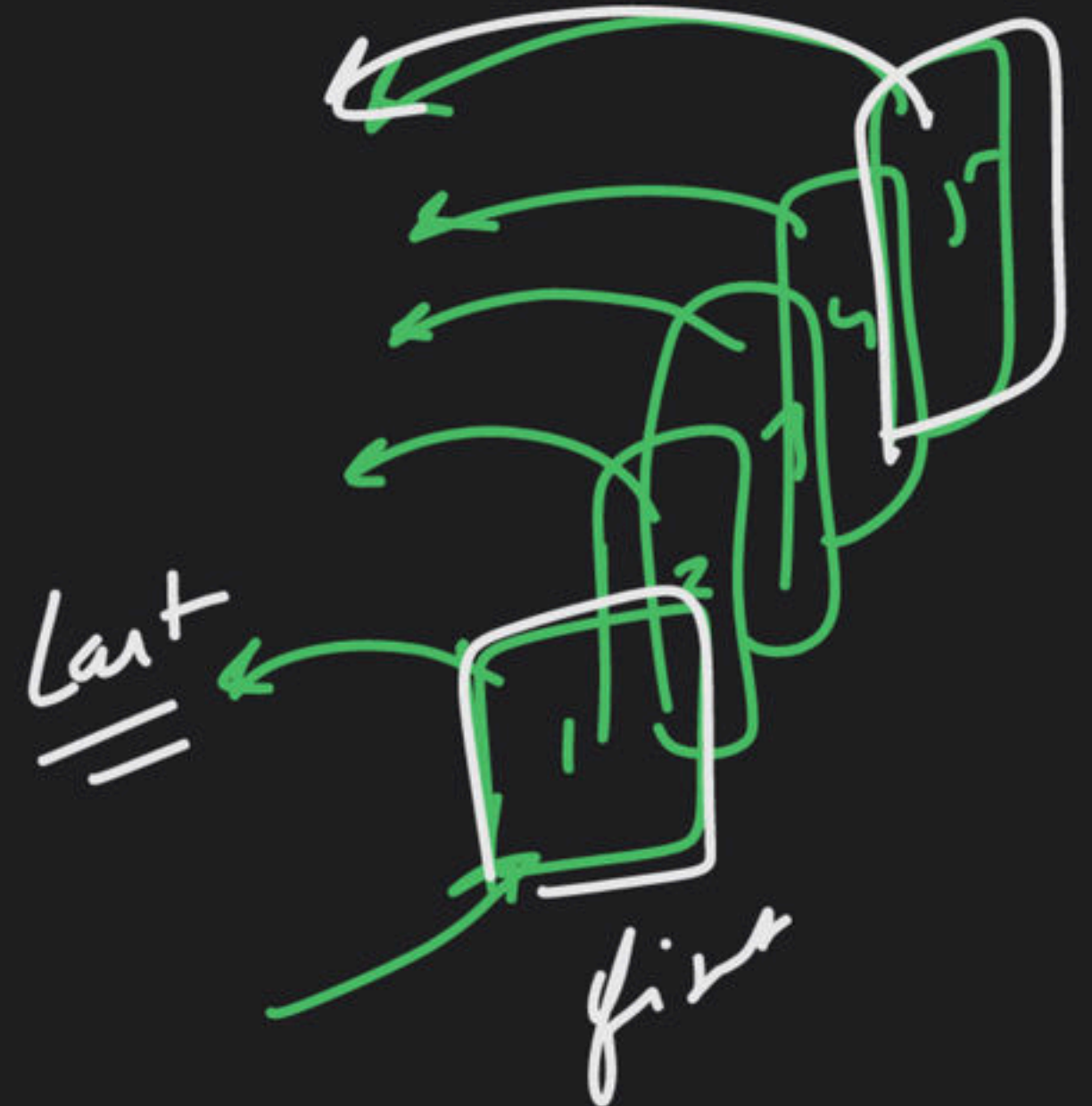
Foundation Course on Data Structures & Algorithms - Part II

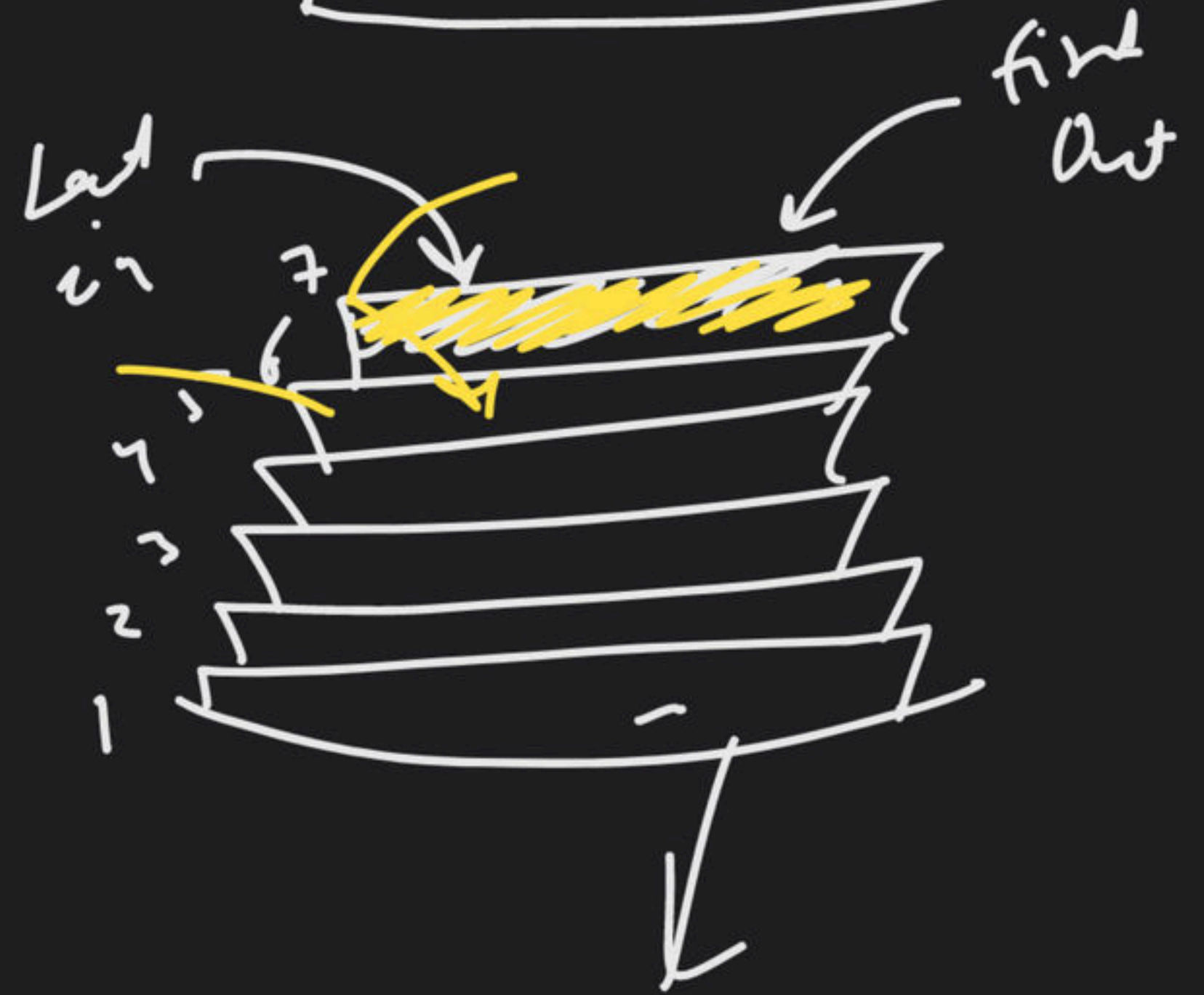
→ Stacks:- (5th class → Unit test)

1, 2, 3, ,



Last in
first out







Stack → is a Data Structure



LIFO

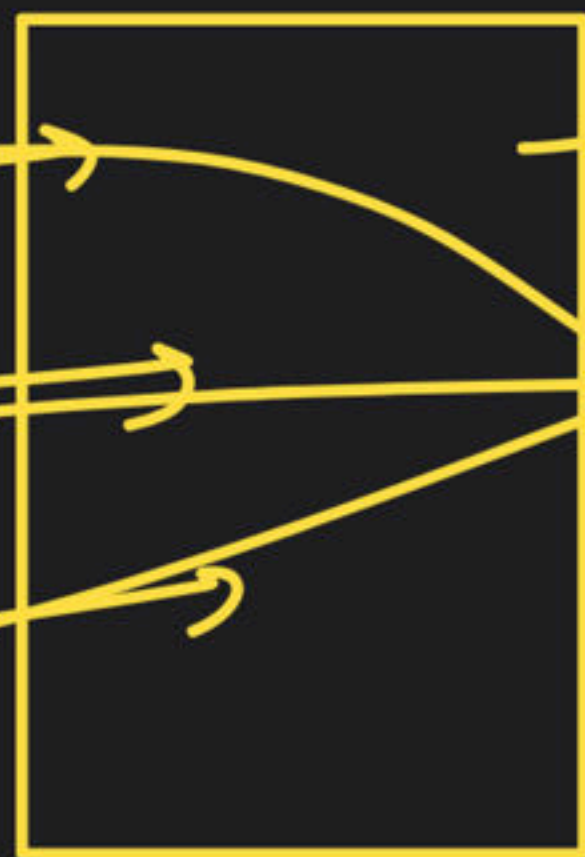
Last in first Out

insertion

(1)

(2)

(3)



removed

3

2

1



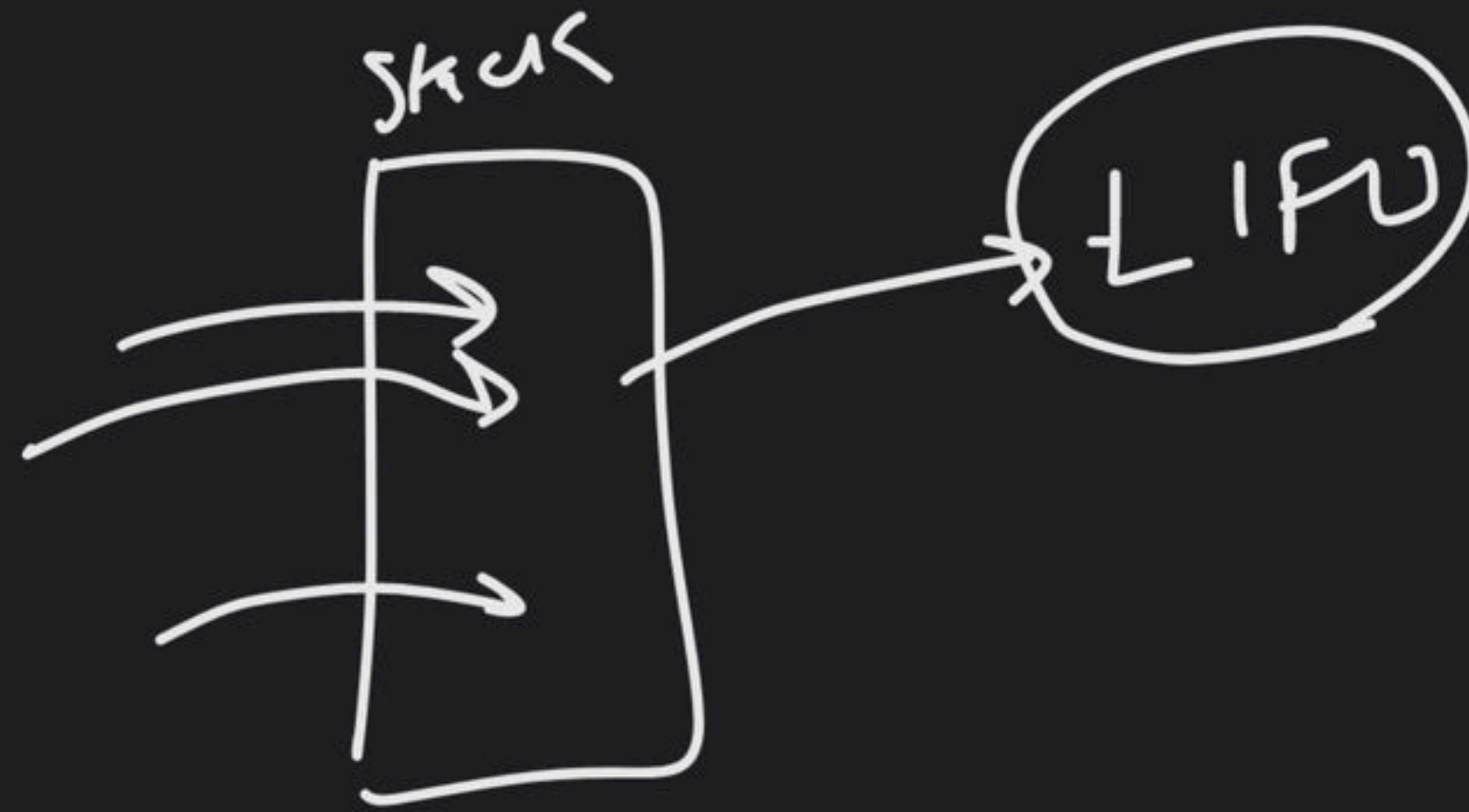
Stack

↓
order → insert

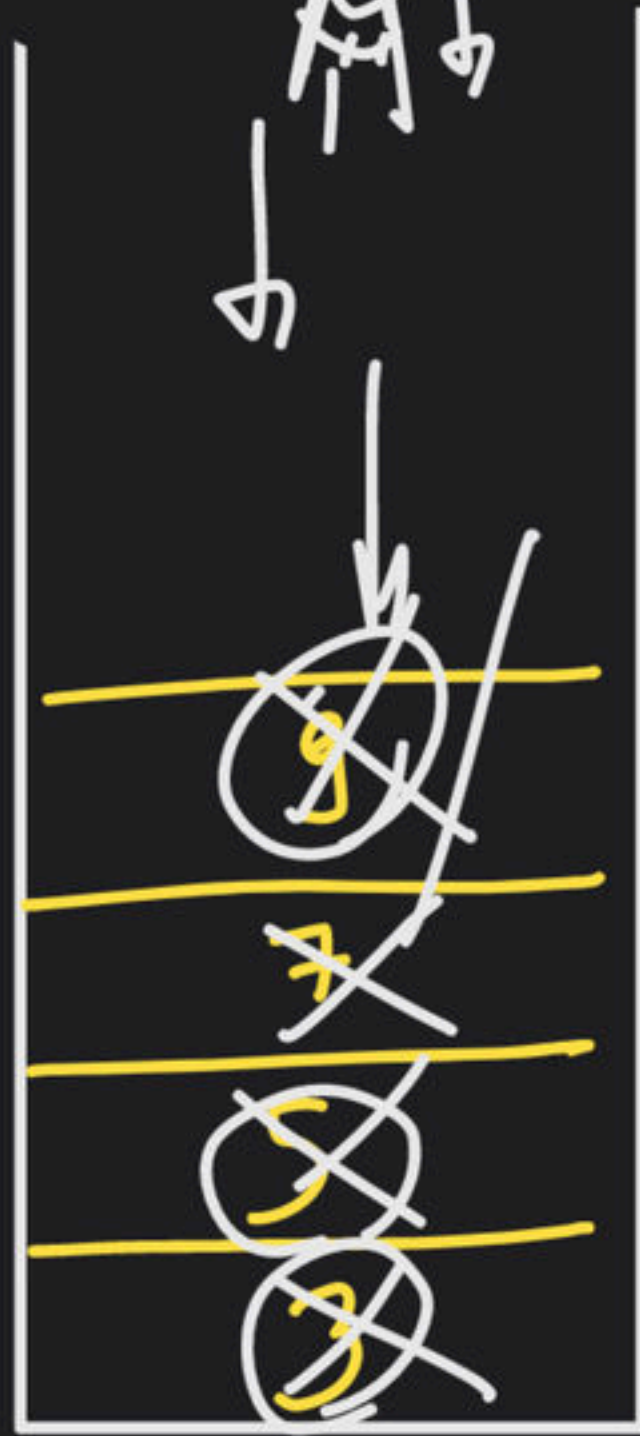
↓
removal

↓
reverse

order



i/p \rightarrow (3), (5), 7, 9



stack

9, 7, 5, 3 \rightarrow reverse

STL

Scratch AC STACK → Implementation

Stack STL

↓

Cppreference.com

↓

Cplusplus.com

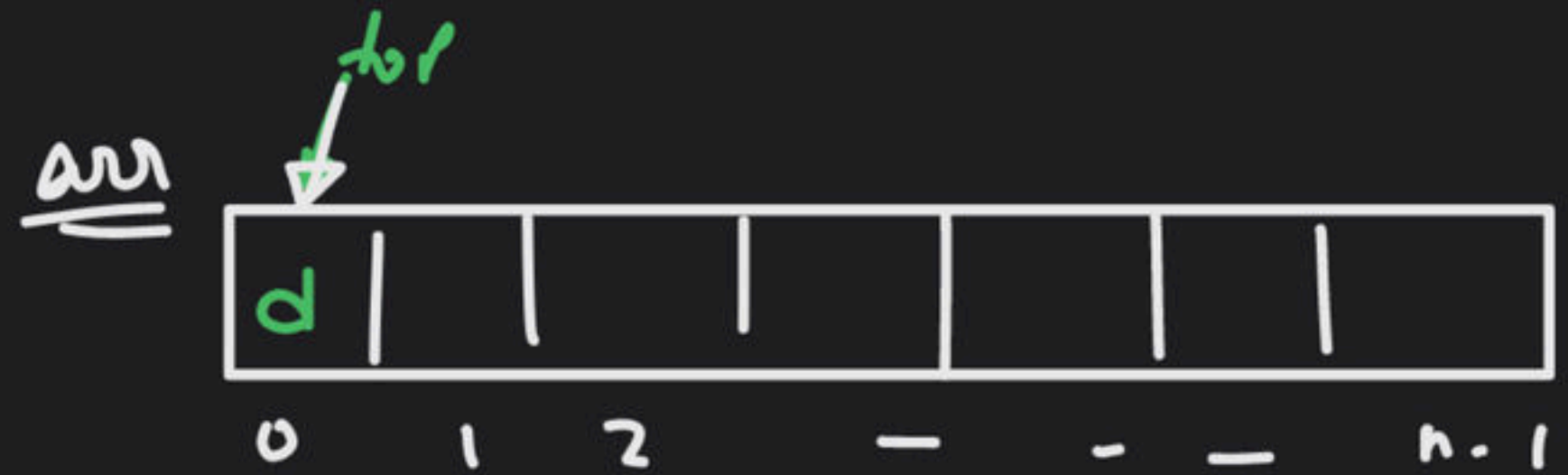

```

class Stack
{

```

push()

};



int top = -1

push (int d)

{ top++;

arr[top] = d;

is Empty ()
if (top == -1)
return true }

pop ()
{ top--;
}

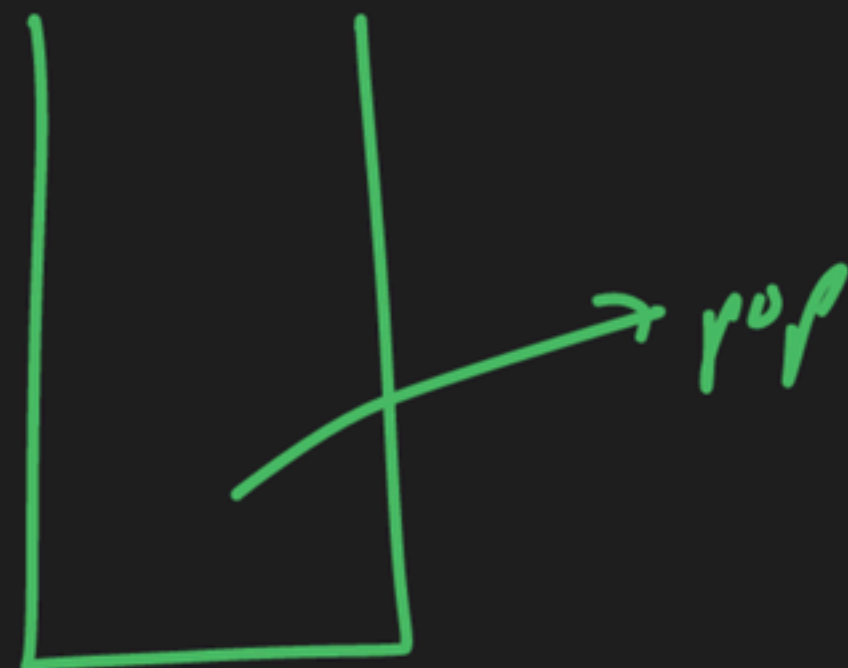
top ()
{ return
arr[top];
}

if ($top = -n - 1$)

or insert nhi kr



arr[5]



if ($top = -1$)

{ cout << "Stack Underflow"

}

Q-1

create a stack using stl;

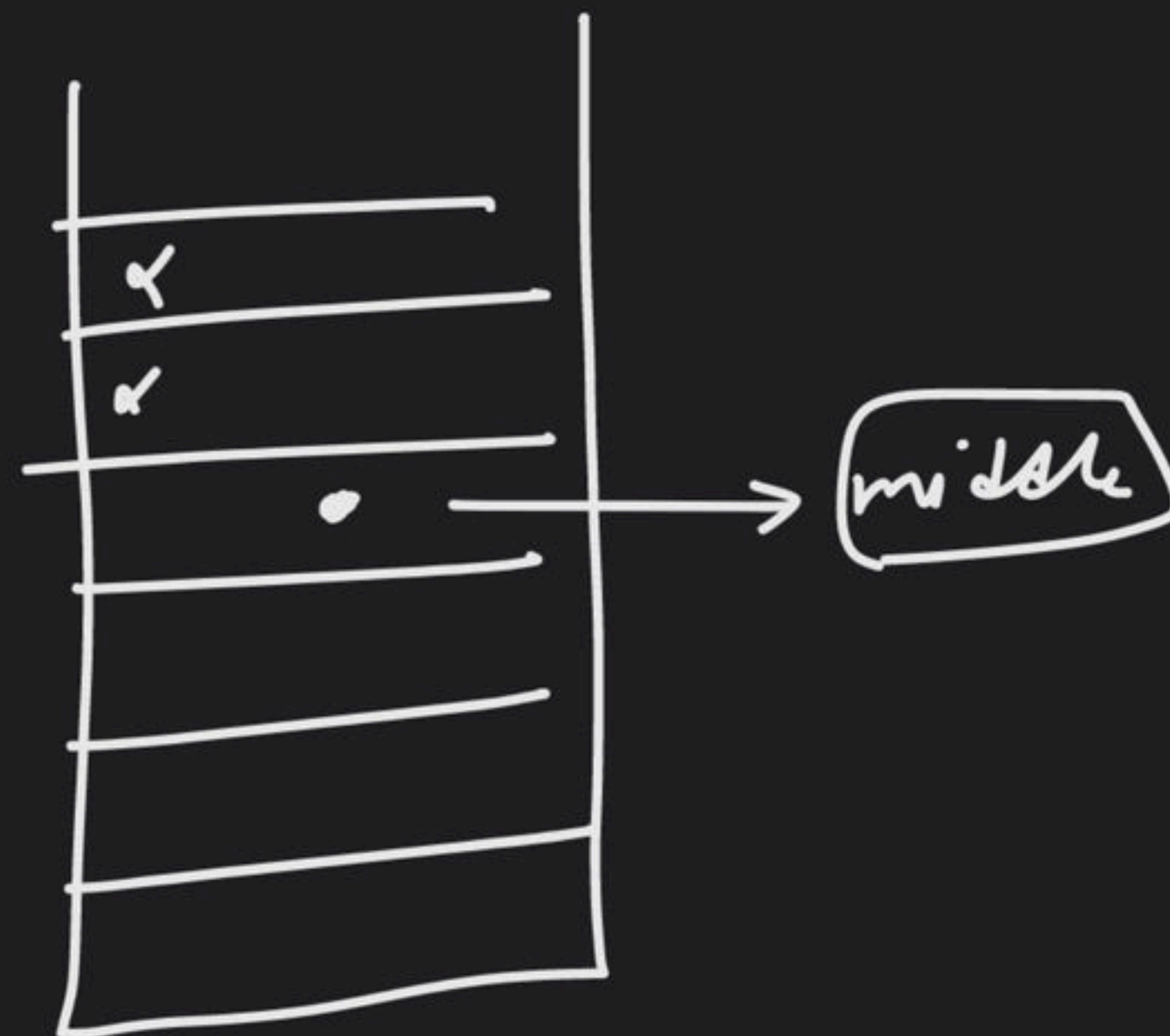
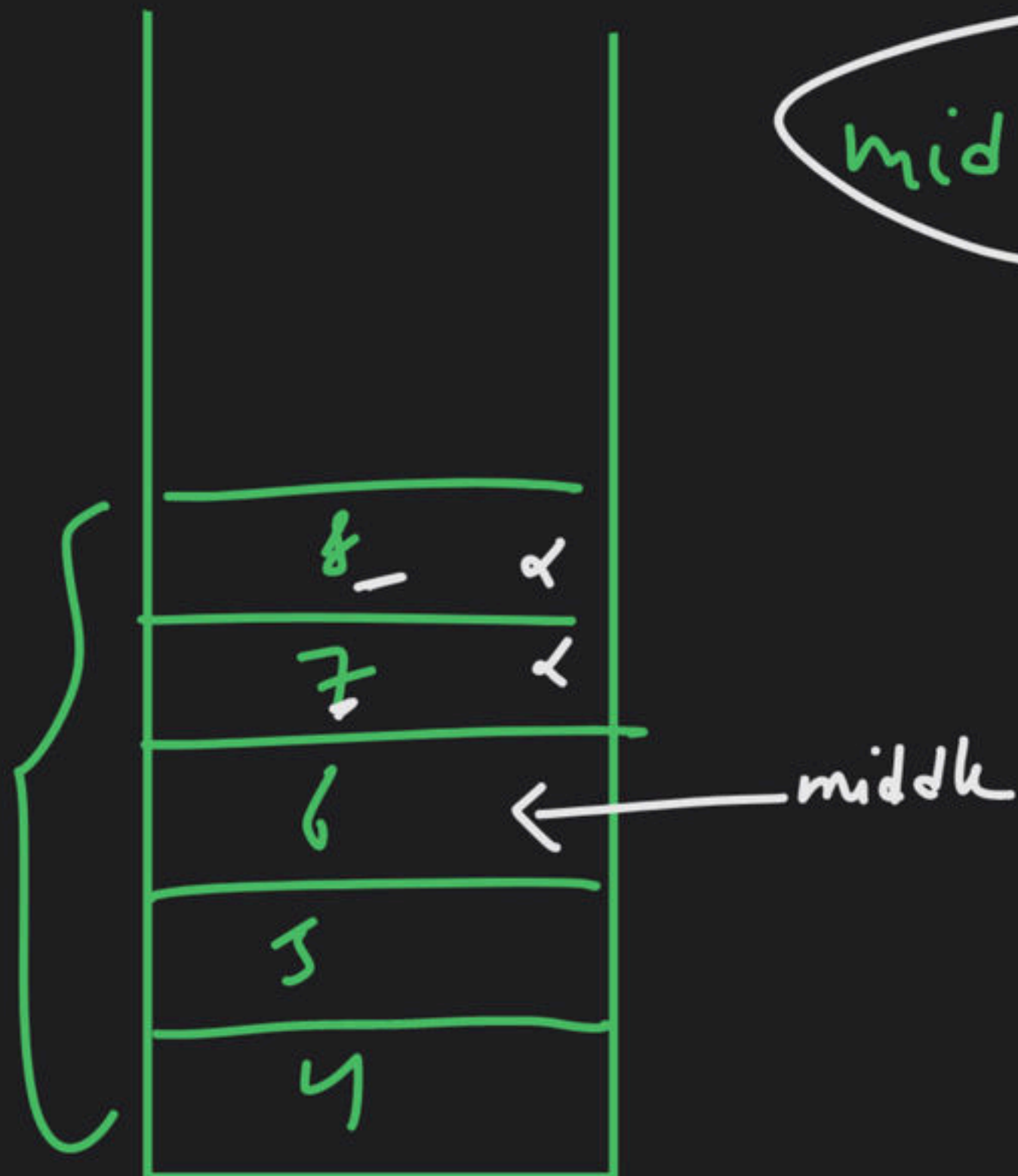
take i/p \rightarrow n

i/p \rightarrow n elements

o/p \rightarrow print all element after removal
from stack

Q →

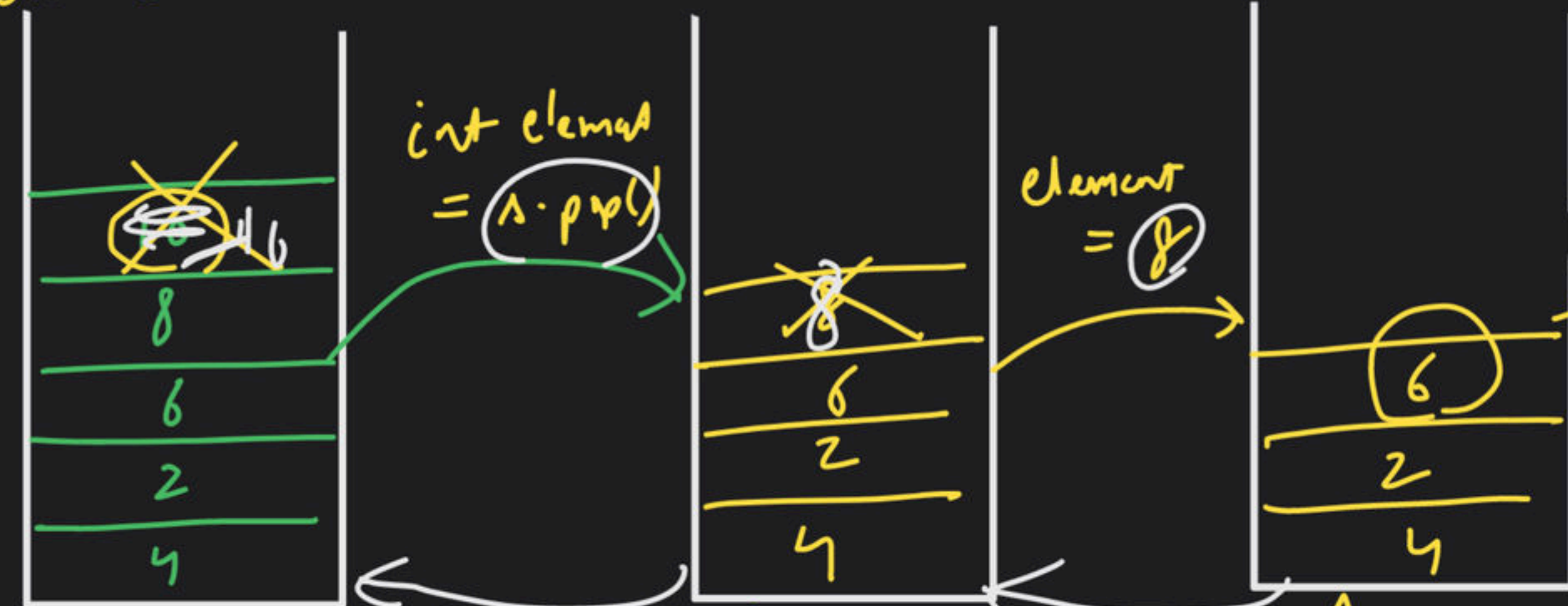
middle element print → ?



Ans →

Total size = 5

Recursion - 1st call



↑
size = 5

←
push(10)

↑
size = 4

←
push(8)

↑
3

size = 3

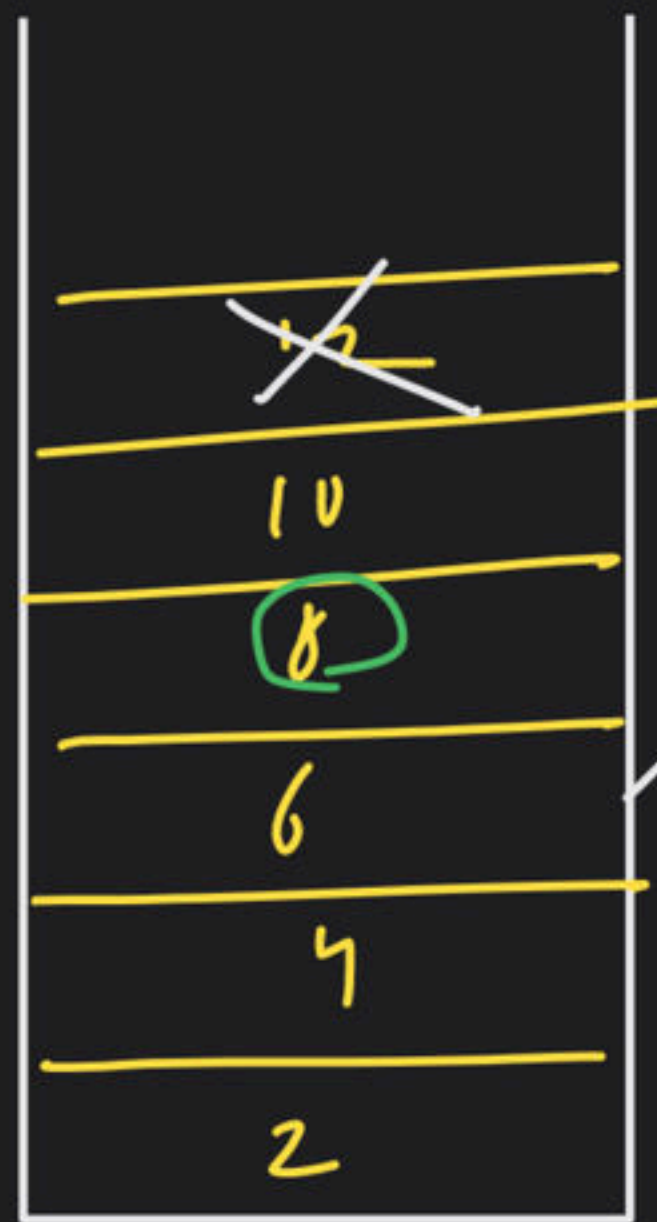
$$\frac{\text{Total size} + 1}{2} = 2$$

$$\frac{\text{Total size} + 1}{2} = \text{size}$$

←
B.C
s.top() → (ans)

Total Size = 6

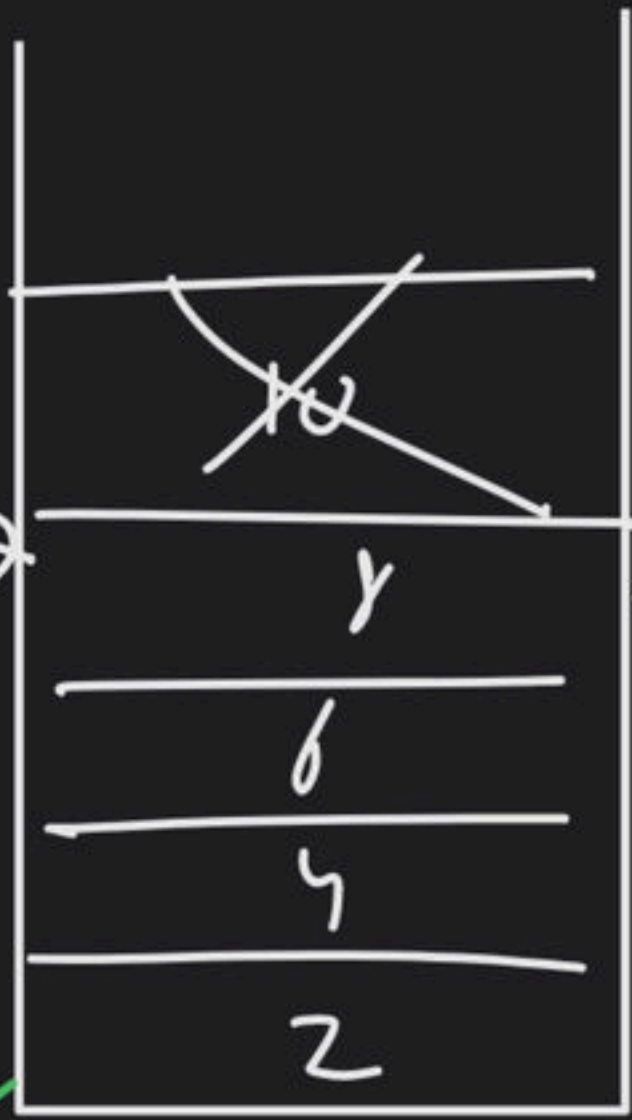
char lahaa hai



I
 $top\ \Sigma\ count = 12$

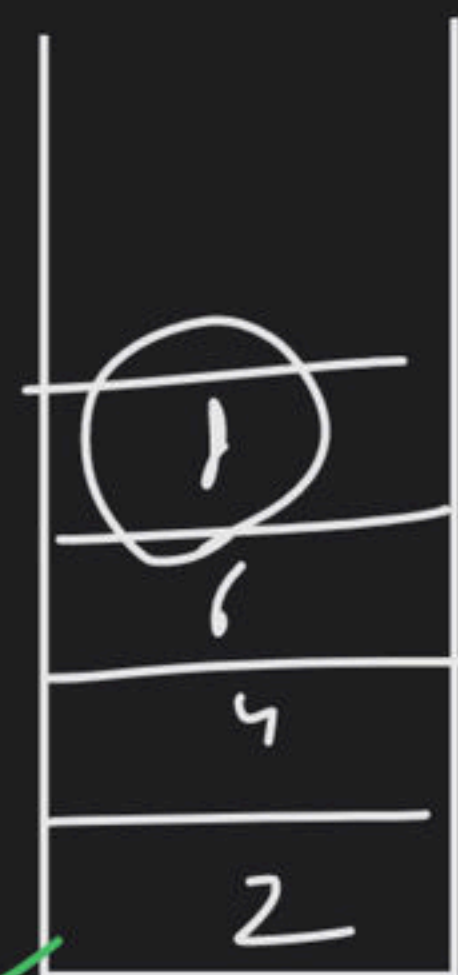
II

III
 $push(12)$



$top\ \Sigma = 10$

$push(10)$



pop
 $size = 4$

$$\frac{Total\ line + 1}{2} = \frac{8}{2}$$

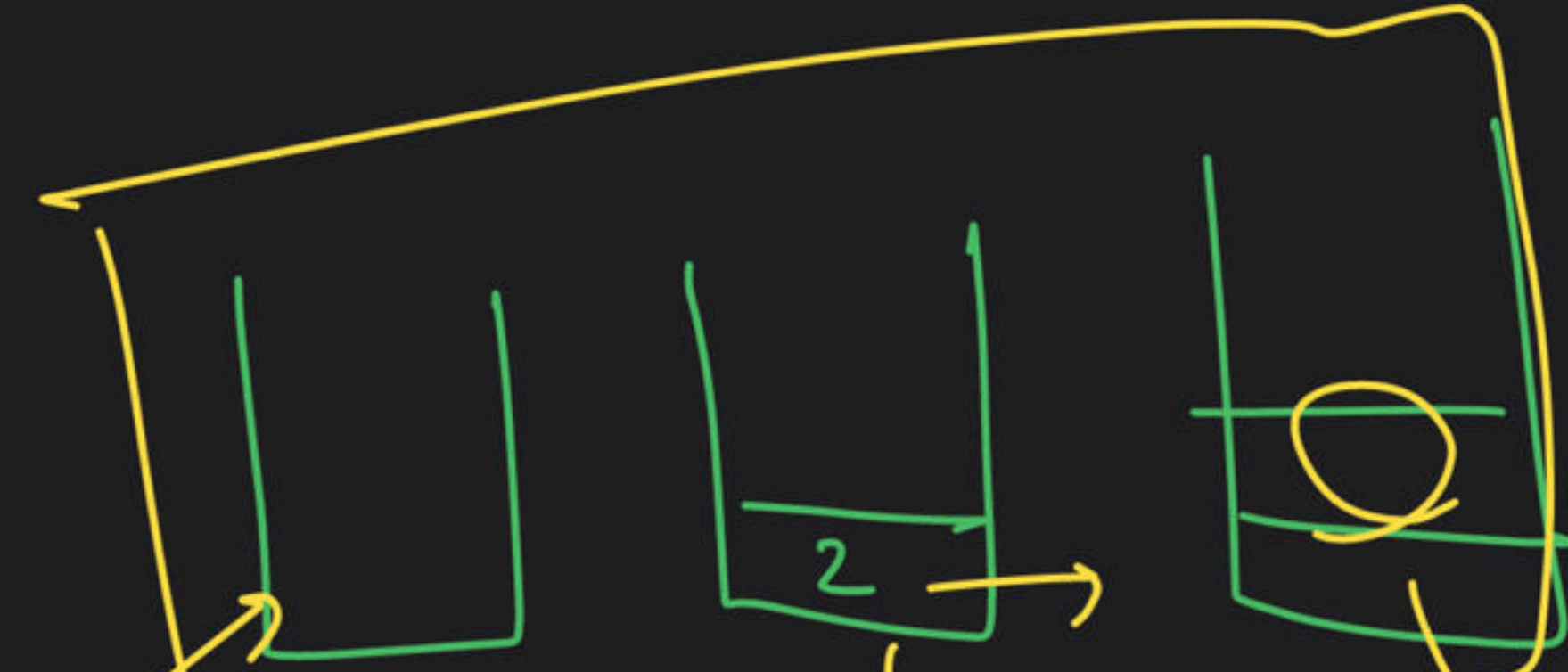
$$= 3 + 1$$

$$= 4$$

$$\frac{Total\ line + 1}{2} = 5$$

$$\therefore top() \rightarrow (8)$$

grait success



$\frac{3}{2} r \sim 2$
 $m \sim 3$



0

$T.S \sim 1$

$\frac{1}{2} r \sim 1$

size ~ 1

$r \sim 1.22$

$\frac{2}{c} r \sim 2$

size ~ 2

\rightarrow
 exp_3

size \rightarrow county \rightarrow back \rightarrow clmt ①

Total size \rightarrow slatig \rightarrow sluk

solve (st, size, TS)

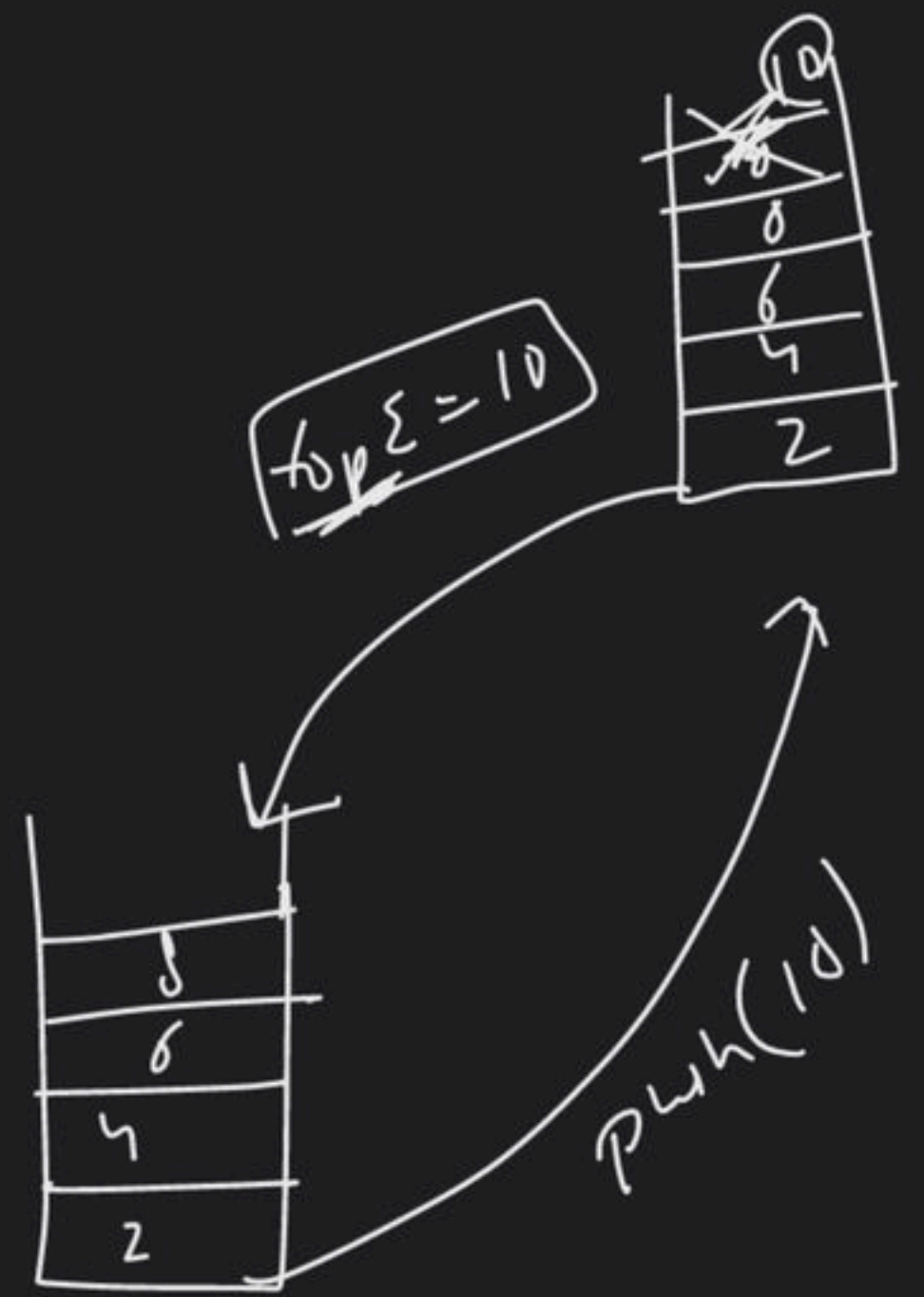
{ if (TS == size)
row
return
int TS = st.top()
st.pop()

solve (st, size+1, TS)
st.push (TC); }

main ()
2

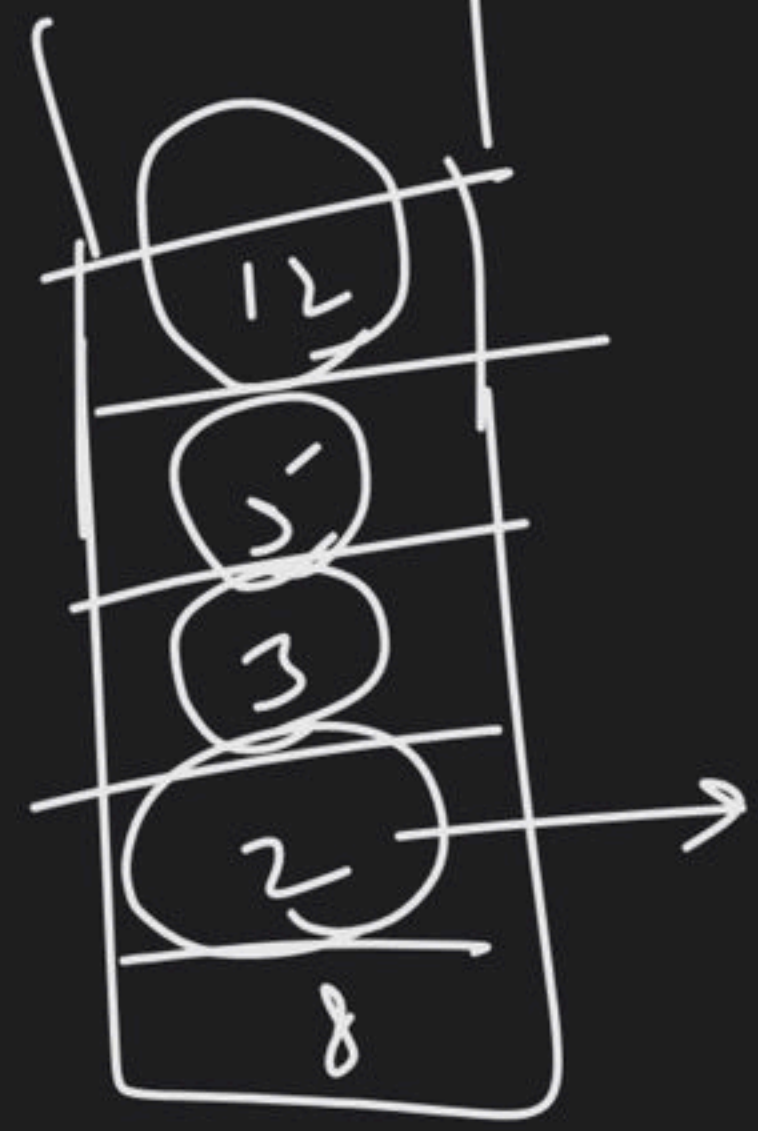
solve (st, 0, $\frac{TS+1}{2}$)
5 \rightarrow ③
6 \rightarrow ④



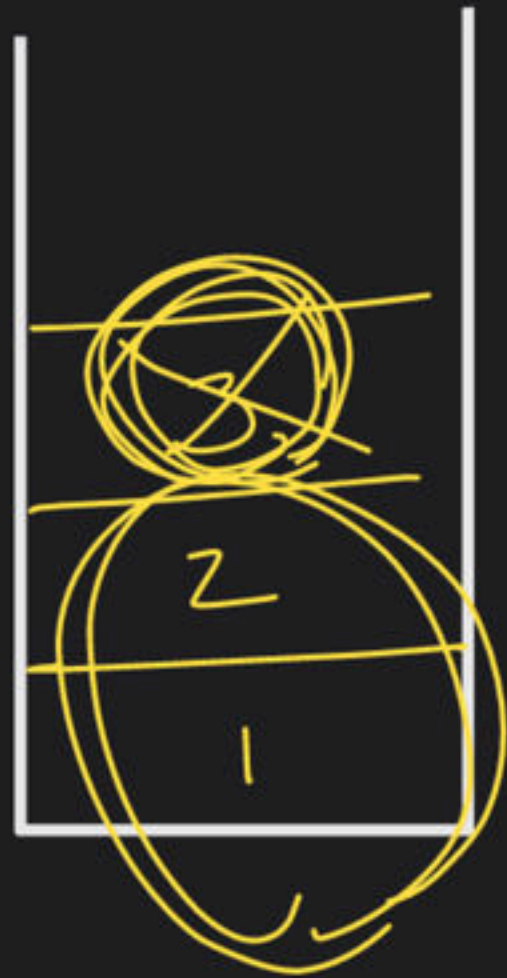


tail Rec

Backtracking



→ Reverse a Stack



(i) $\text{topElement} = \text{st.top}()$
 st.pop

(ii) $\text{reverse}(\text{st})$

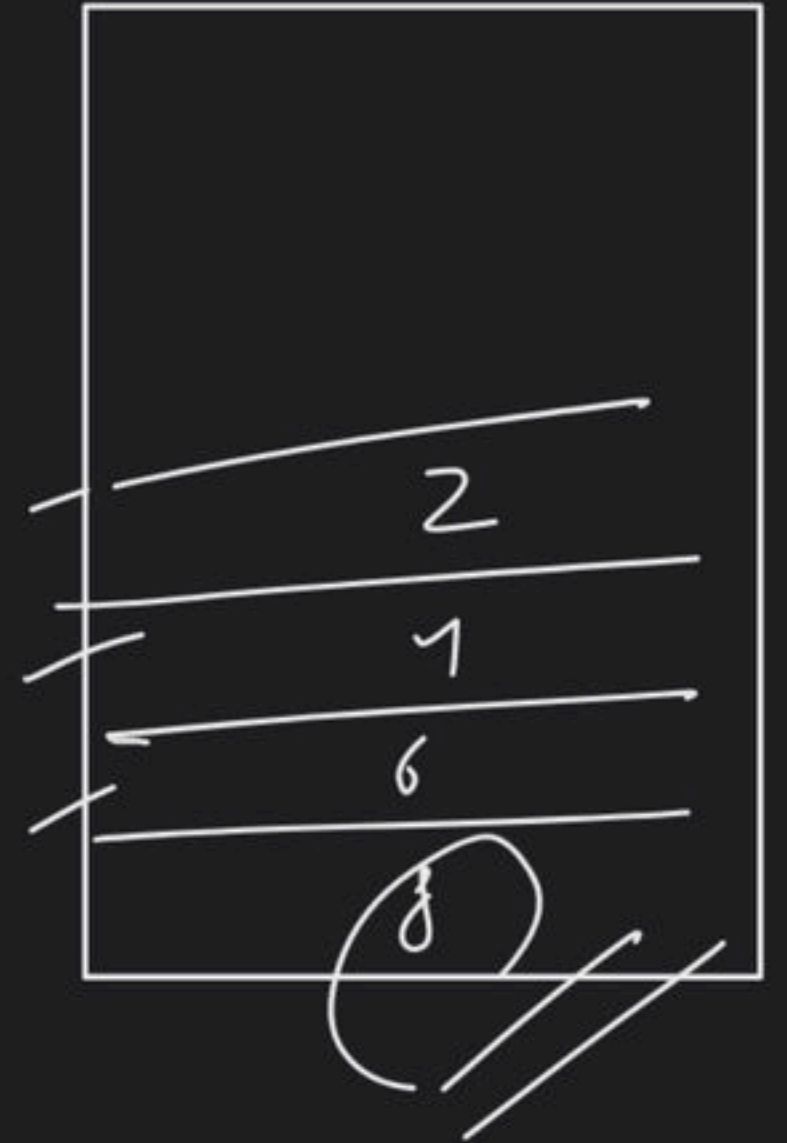
(iii) $\text{insertATBottom}(\text{st}, \text{topElement})$



(1) $int \text{ topElement} = st.top()$
 $st.pop()$

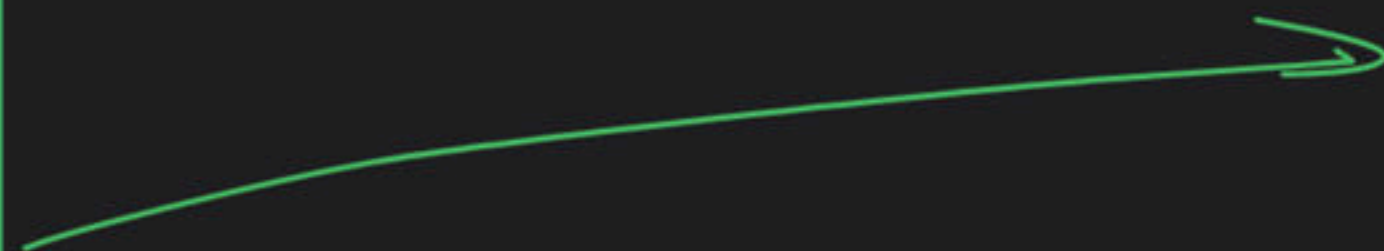
(2) $reverse(st)$

(3) insertAtBottom(st , $topElement$),



→ Q → insert At Bottom

7
3
2
1



3
2
1
7

8
6
7
2

element = 8

6
7
2
8

$tE = 6$

7
2
8

$tE = 7$

2
8

$tE = 2$

8

13 c

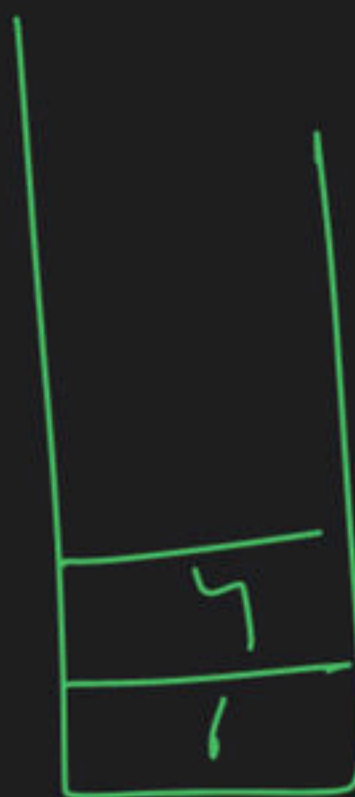
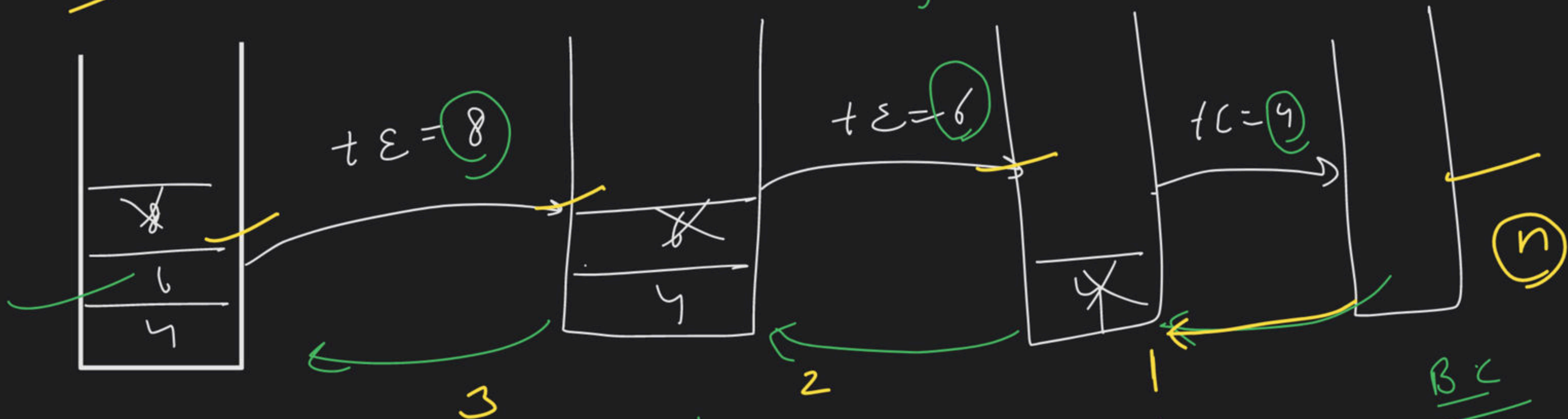
push(6)

push(7)

push(2)

Recursion

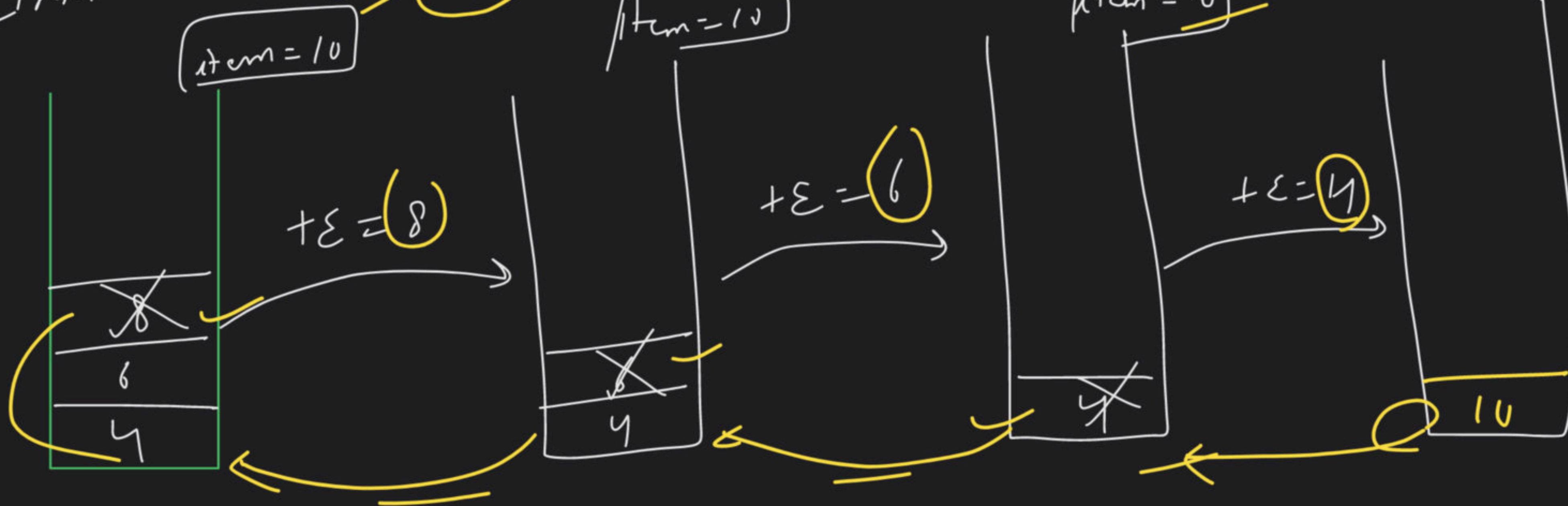
Majoraki is



n^2

S.C

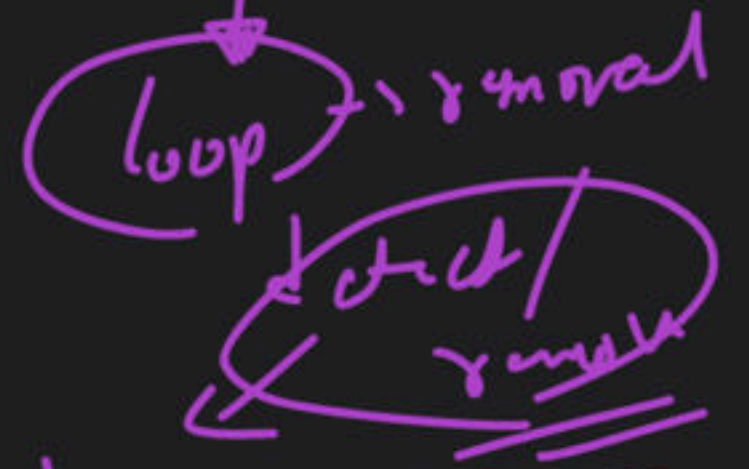
Insert At Bottom T-C



$O(n)$

100% (an, an +)
 Interview Experience
 reject ← [D.E show]

Ladki → necklace



07

14 → Aptitude

Week

GFG

10 MT

Trained

Li

11 ~ 12/7

3 → coding

1 Brute force → 1087

15 → ptr

Solve huc ya nahi - ?



```

int solve (vector<int> &arr)
{
  if (arr.size() == 3)
    return -1;
  return solve(arr) + 2;
}
  
```

cout << solve (arr)



Student

Stack Overflow

2nd Question

Detect/ delete
loop

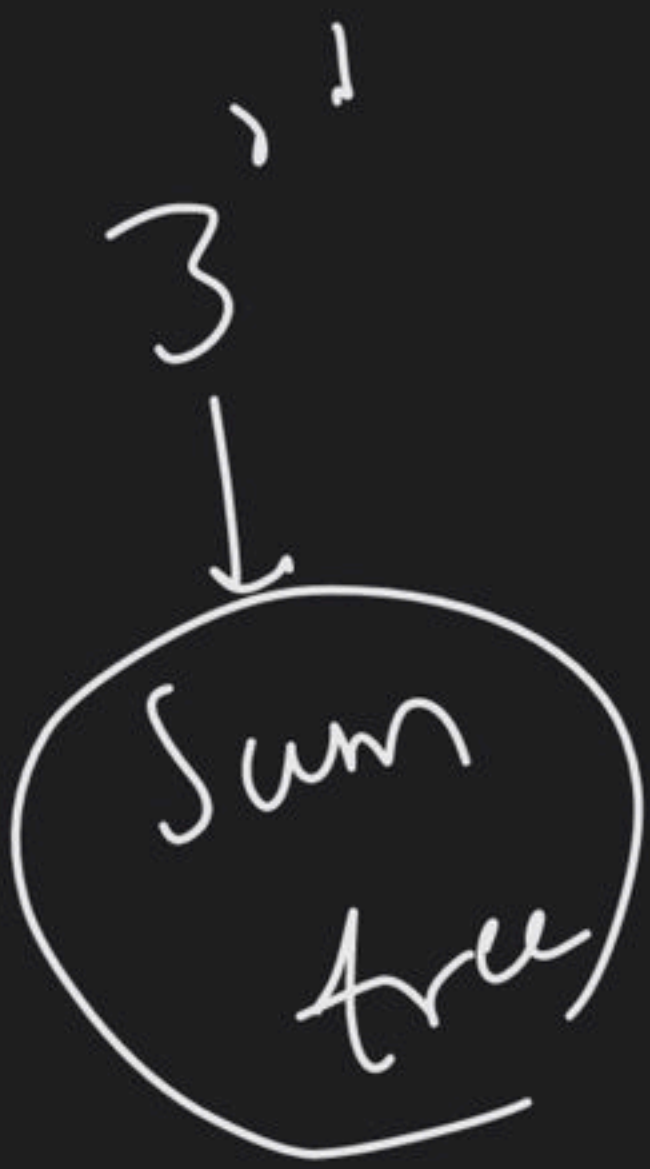
1 hr

1 | 2 | 3 | 1

1 | 2 | 3 | 1

1 | 2 | 3 | 1

1 | 2 | 3 | 4



```
int main()  
{
```

```
int *ptr = 23;
```

valid or not?

30

```
int **ptr2 = &ptr;
```

min

error
↓
why?

```
*ptr += 5;
```

```
**ptr2 = *ptr + 2;
```

```
cout << **ptr2;  
}
```



1st Round

3 9

Intro

Binary Search algorithm

Rec

Fractional Knapsack

greedy algo

Sorting

did old age -> phone

Wildcard Pattern

Rec + DP

Numeric Keypair

Question

Matching

BT

H/W

YT

OS -> Page fault -> ?
Paging -> ?
Dining Philosopher

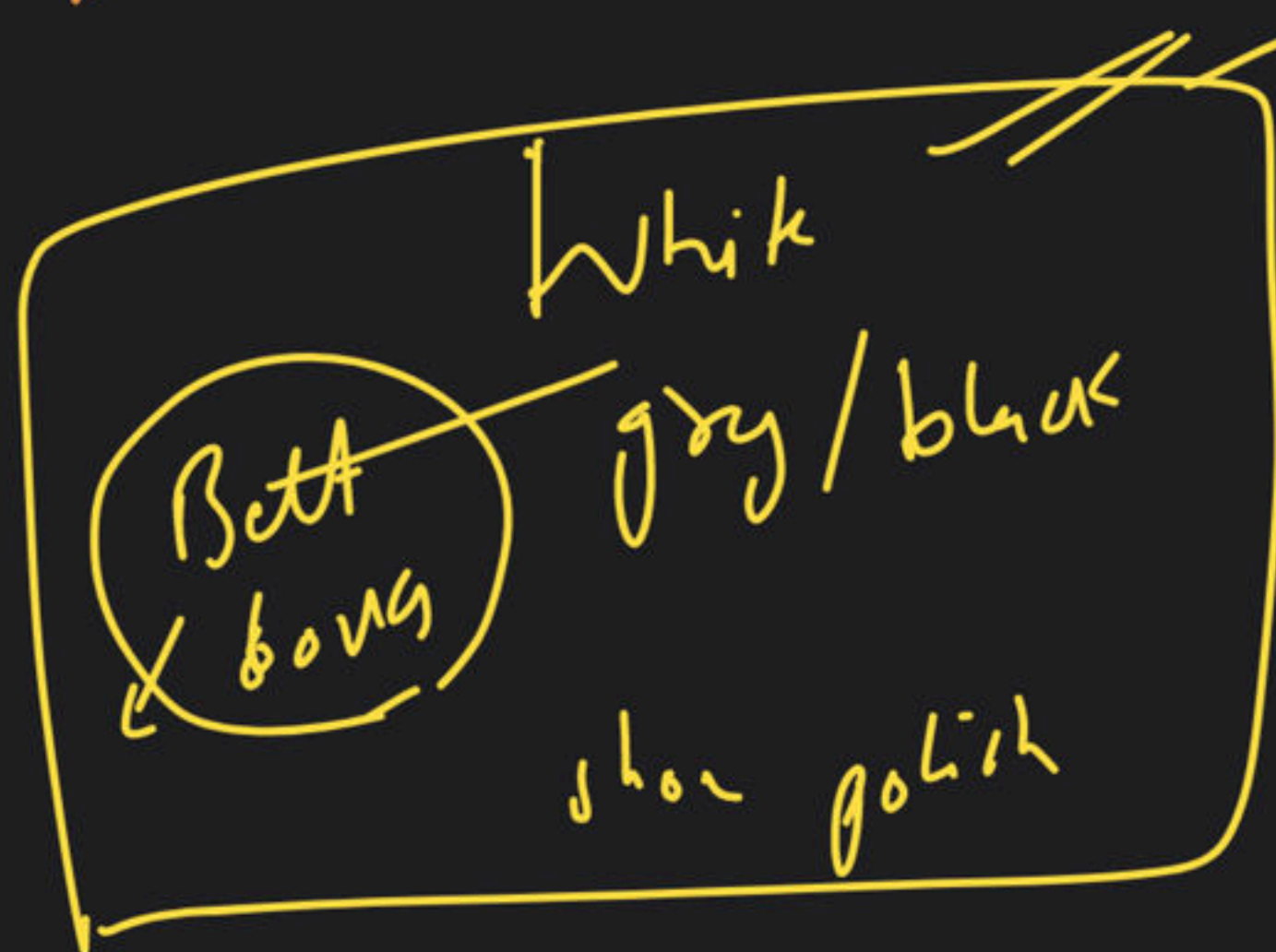
SQL -> ER Model

Sum tree

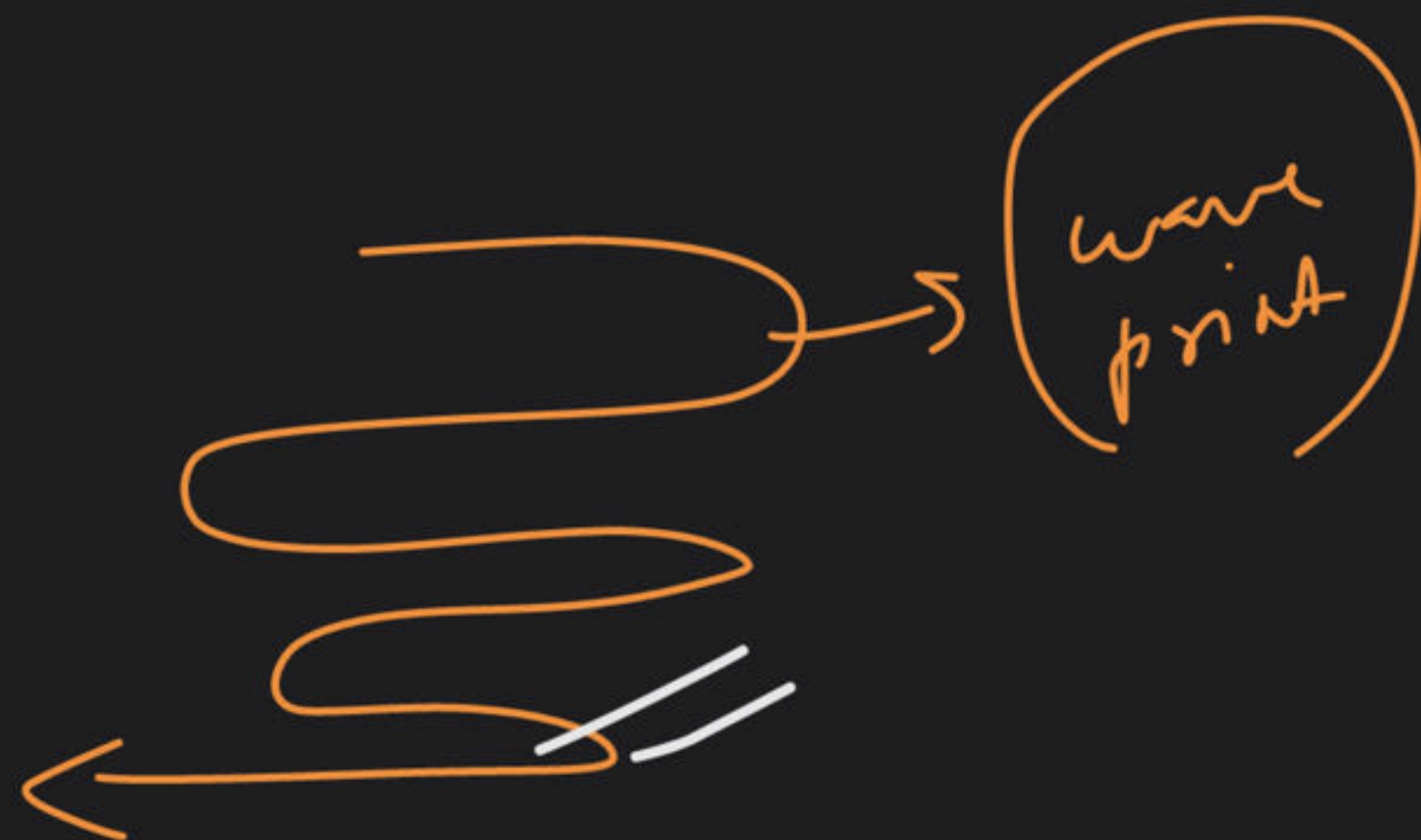
1 -> a/c
2 -> d/c
3 -> g/c

Snake Pattern

Blue - X



Spiral
Print



Graph \rightarrow

Dijkstra's \rightarrow

MR

Boost
up

H/w

y^x
 \downarrow
 z^x

mehnat
karu

→ Restaurant ← Website

ORB

LC
→ Google ↑

Darna
nahi
hai

Motivation

→ Data mat
Karo

1.5 hr

2.5 hr