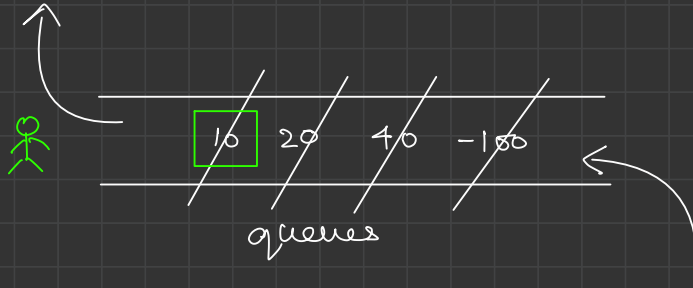




Queues

→ linear data structure



add

- | | | |
|---|------|---|
| ① | 10 | ✓ |
| ② | 20 | ✓ |
| ③ | 40 | |
| ④ | -100 | |
- ↓

remove

- | | | |
|---|------|---|
| ① | 10 | ✓ |
| ② | 20 | ✓ |
| ③ | 40 | |
| ④ | -100 | |
- ↓

first in first out (FIFO)

Stacks

→ follows last in first out (LIFO)

→ Methods:

- | | | |
|--------|----------|---|
| ① push | Tc : 001 | } |
| ② pop | Tc : 001 | |
| ③ peek | Tc : 001 | |
| ④ size | Tc : 001 | |



enqueue

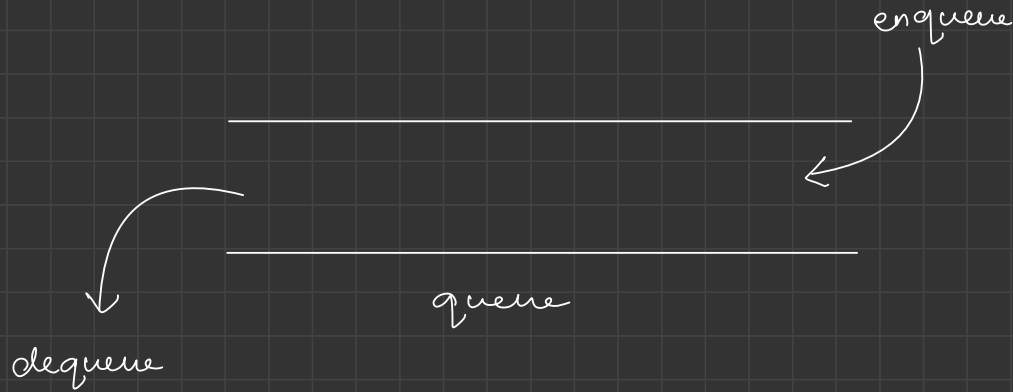
→ Adding element in a queue

Tc.
 $O(1)$

dequeue

→ removing element from a queue

Tc.
 $O(1)$



Methods of Queues

① add()

enqueue

② remove()

dequeue

③ peek()

to see the front
ele.

④ size()

no. of ele.

TC: O(1)

Queue <G> que_name = new ArrayDeque <>();

Queue <G> que_name = new LinkedList <>();

① offer()

② poll()

③ front()

④ size()

queue

↳ linear DS

↳ follows fifo (first in, first out)

Methods

↳ enqueue, dequeue, peek(), size()

↓

add()

↓

remove()

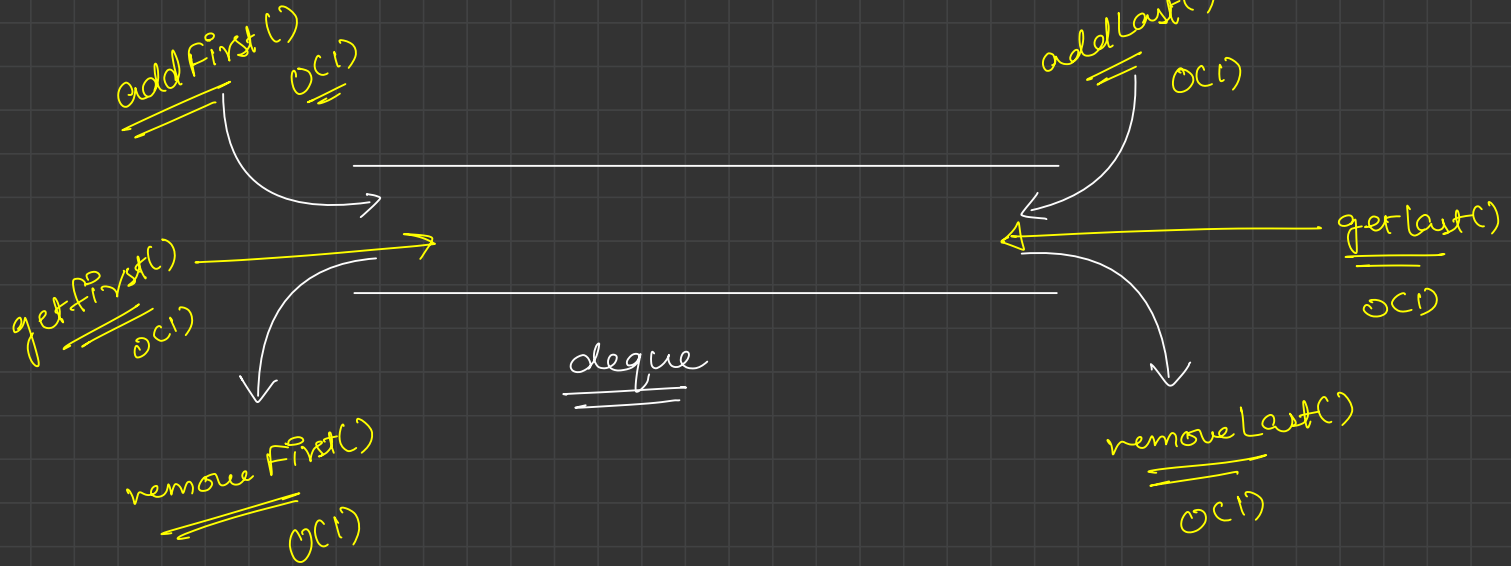
Deque



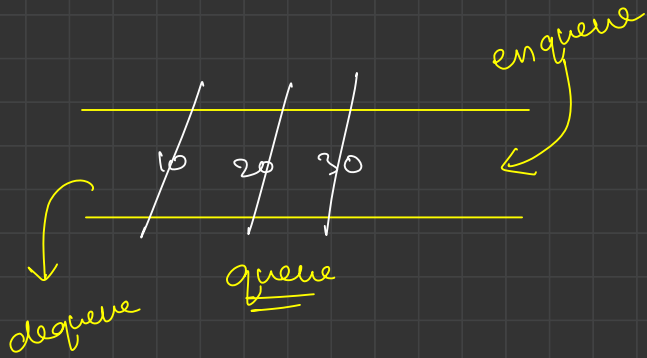
doubly ended queue

↳ implement using doubly linkedlist internally.

Linear data Structure



Q Can you implement a queue using a Deque?

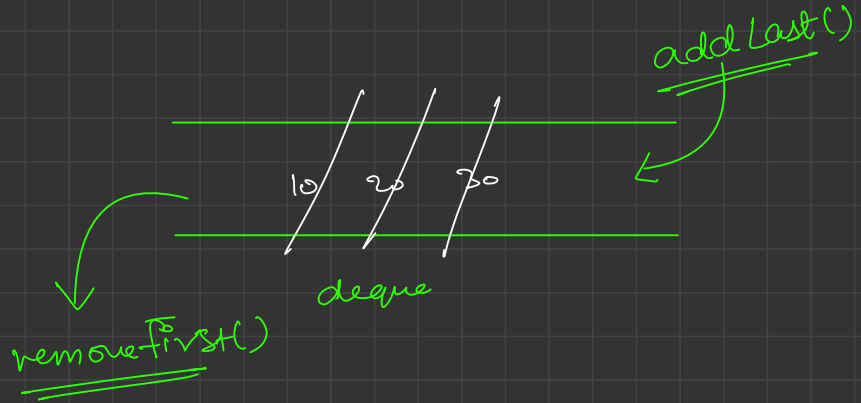


Add

- ① 10
- ② 20
- ③ 30

remove

10
20
30 ↓

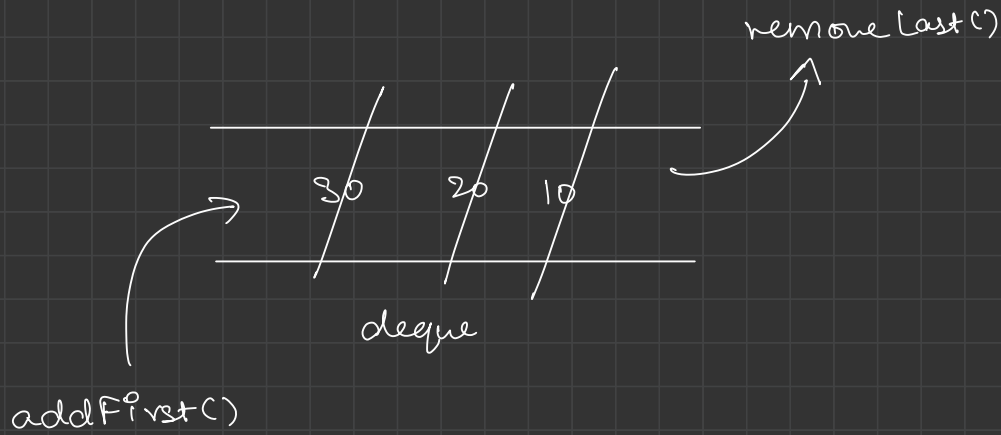


addLast()

10
20
30

removeFirst()

10
20
30



addFirst()

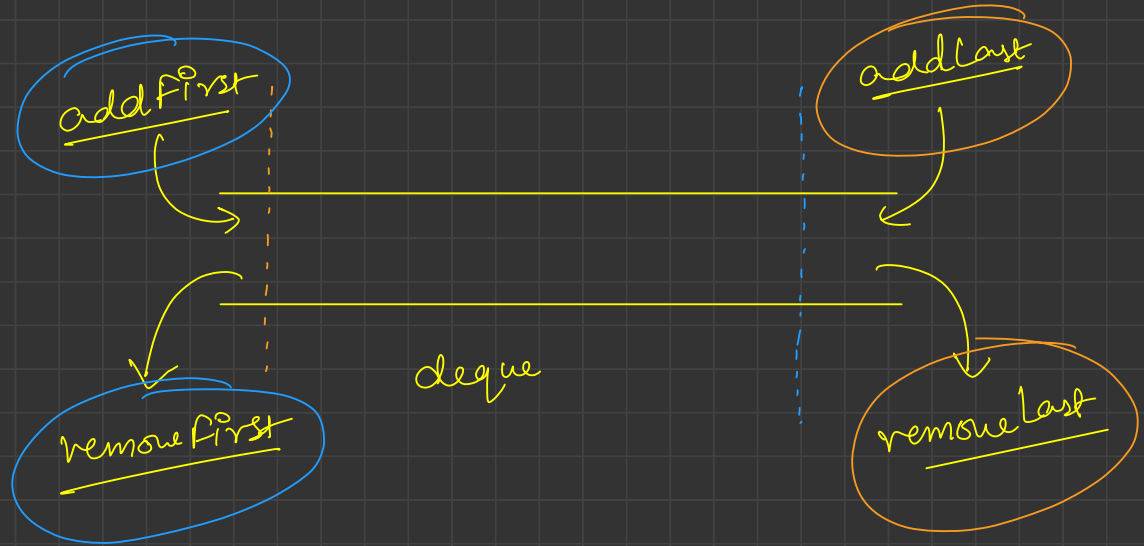
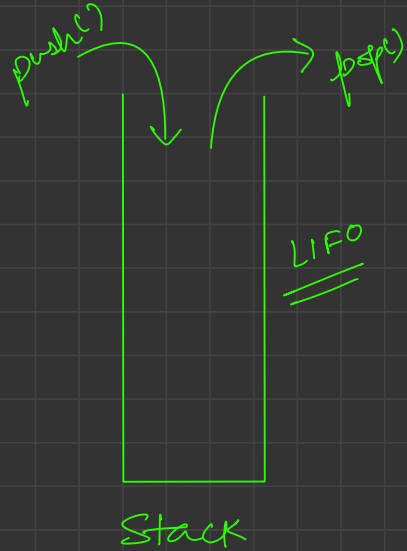
10
20
30

removeLast()

10
20
30

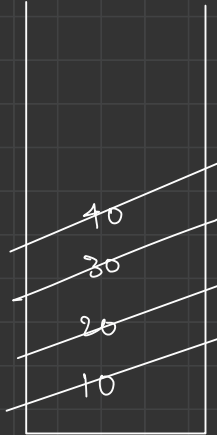
(FIFO)

Q Can you implement a stack using deque?



Q Design a stack using linkedlist

✓ push(10)
✓ push(20)
✓ push(30)
✓ pop() → 30
push(40)
pop() → 40
pop() → 20
pop() → 10



Stack

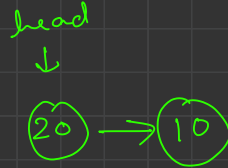
X



TC: O(N)

tail = null

{
curr = head;
while (curr.next != tail)
curr = curr.next;
}



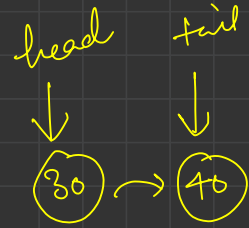
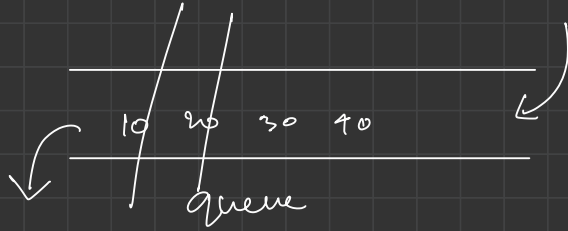
addFirst()
→ of linkedlist

push O(1)

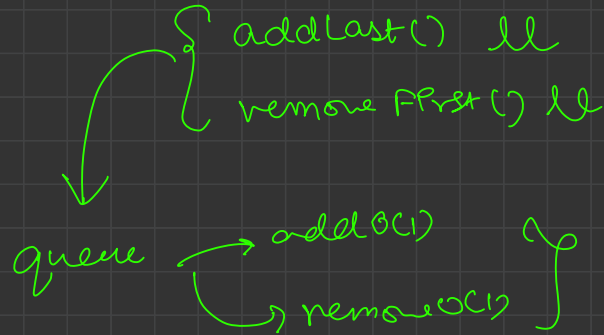
removeFirst()
→ of ll

pop O(1)

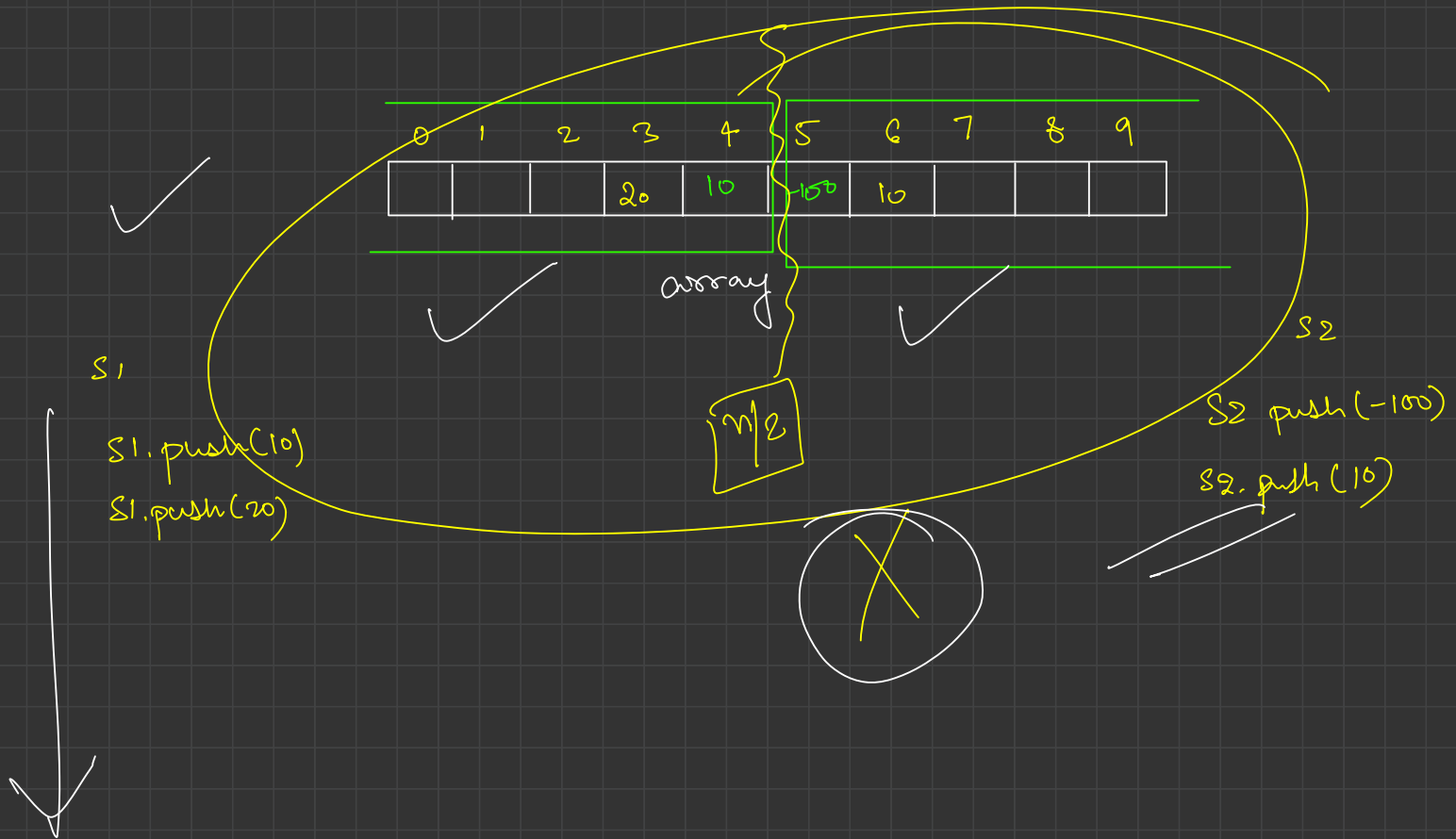
Q implement queue using linkedlist.

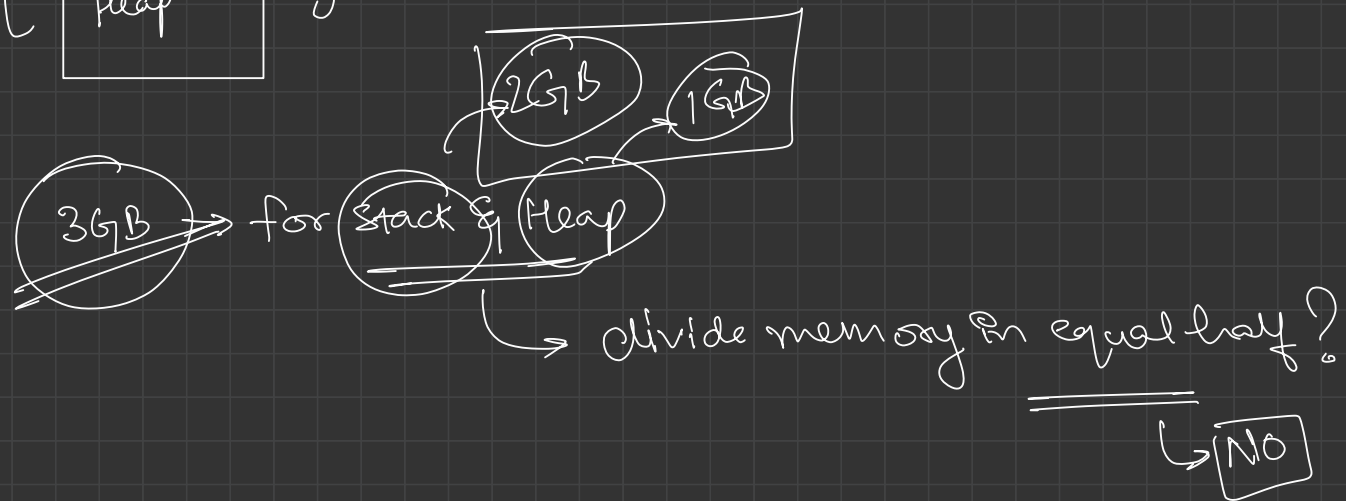
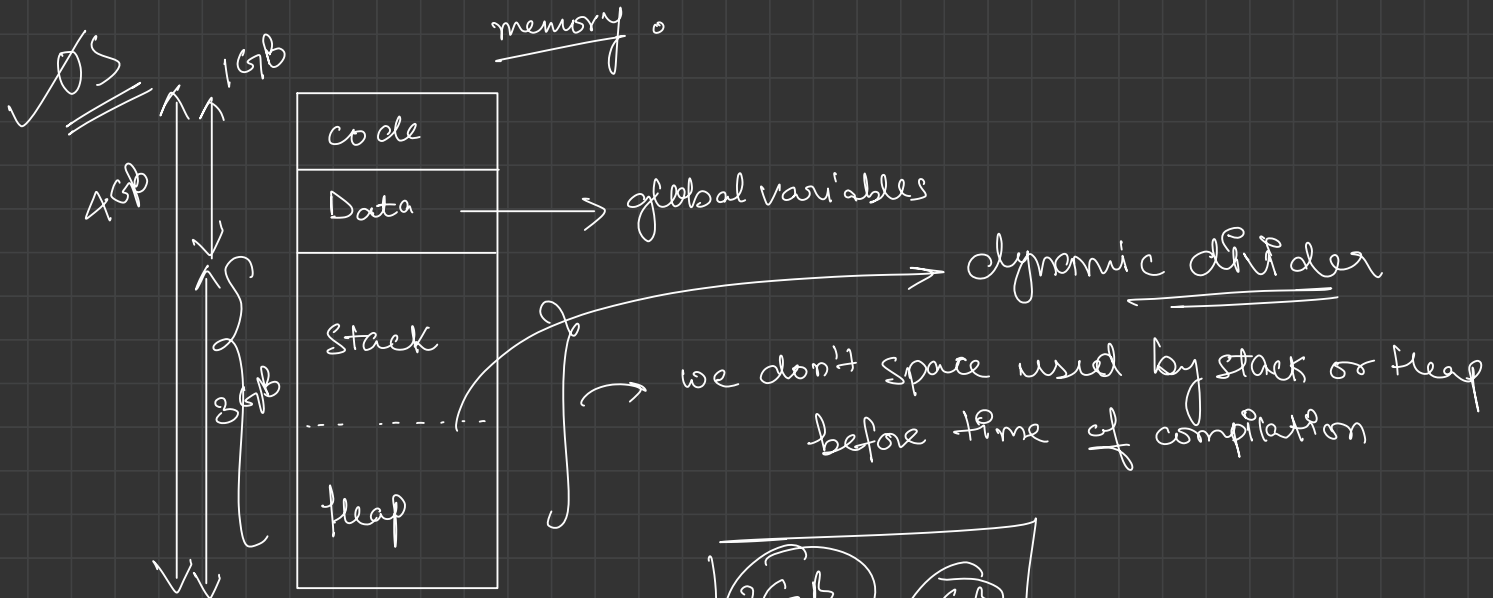


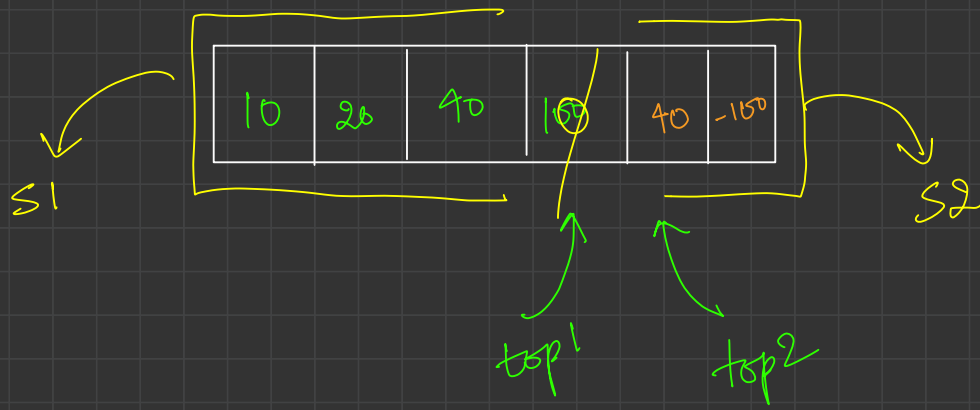
- ✓ add(10)
- ✓ add(20)
- ✓ add(30)
- ✓ add(40)
- ✓ remove() → 10
- ✓ remove() → 20



Q Implement two stacks using a single arr array!





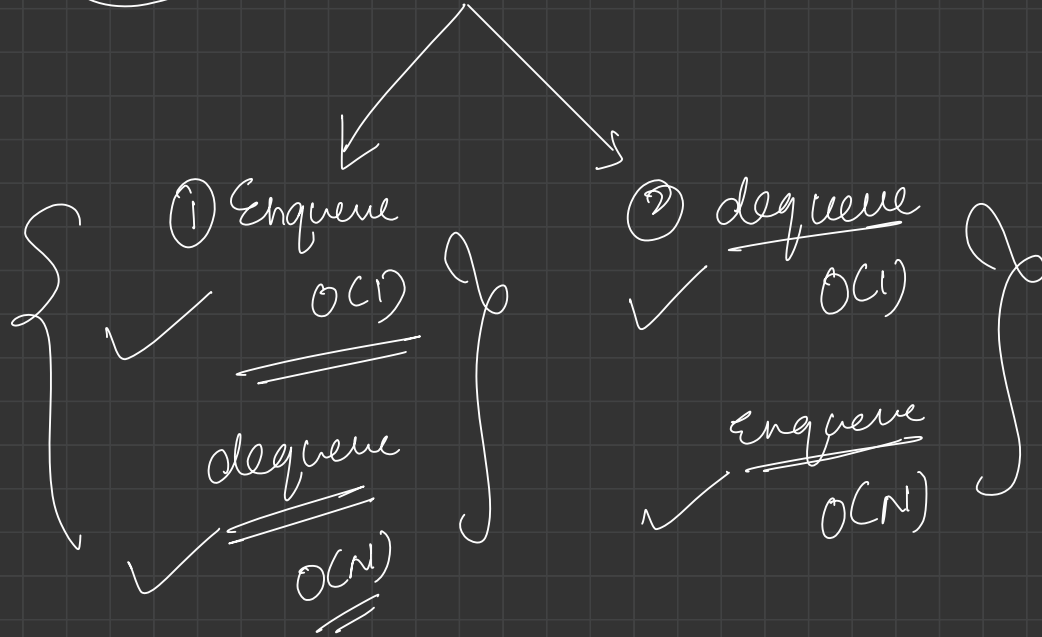


S1.push(10)
S1.push(20)
S1.push(40)

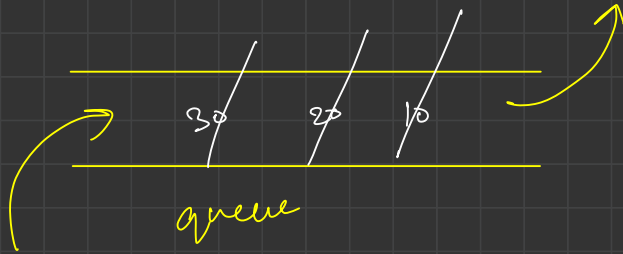
→ S1.push(100)
S2.push(-100)
S2.push(40)

crash → S2.push(40)

Q Implement Queue using stacks (2 stack)



② Deque $O(1)$

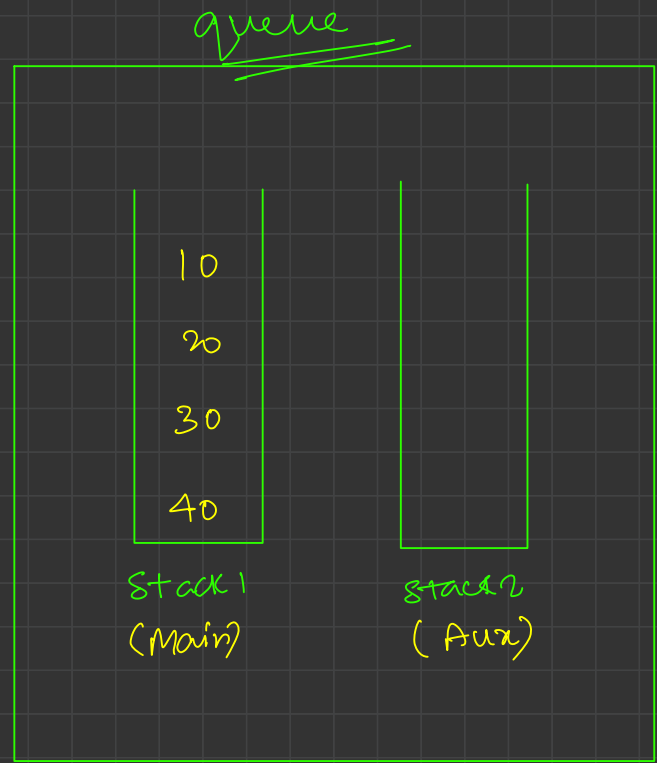


add
10
20
30
40

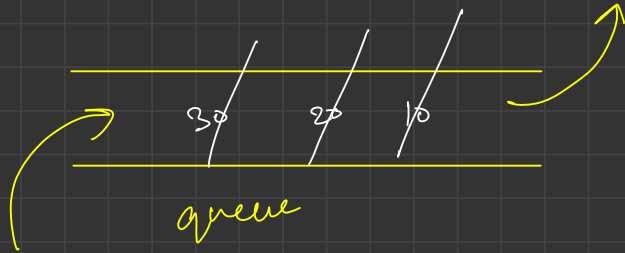
remove
10
20
30
↓

$O(1)$
 $O(N)$

dequeue → pop() stack 1
enqueue → add ele to bottom of stack 1



① Enqueue $O(1)$



add

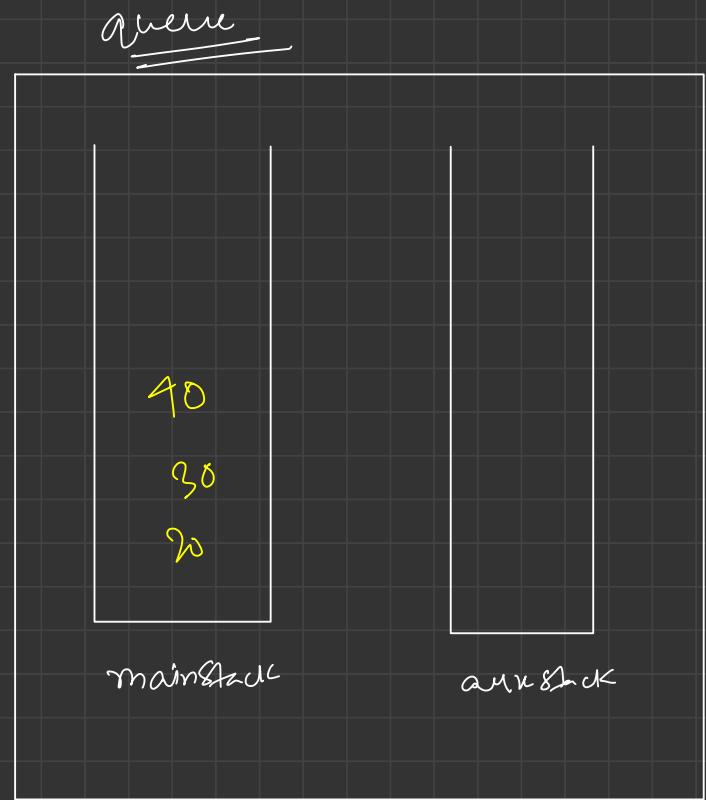
10
20
30
40

remove

10
20
30
40

↓

10



Enqueue \rightarrow push mainStack $O(1)$
dequeue \rightarrow get bottommost Ele of mainStack $O(N)$

add

✓ 10 ✓

✓ 20

✓ 30

✓ 40

✓ 50 ✓

✓ 60

remove

✓ 10

20

30

40

50

60

