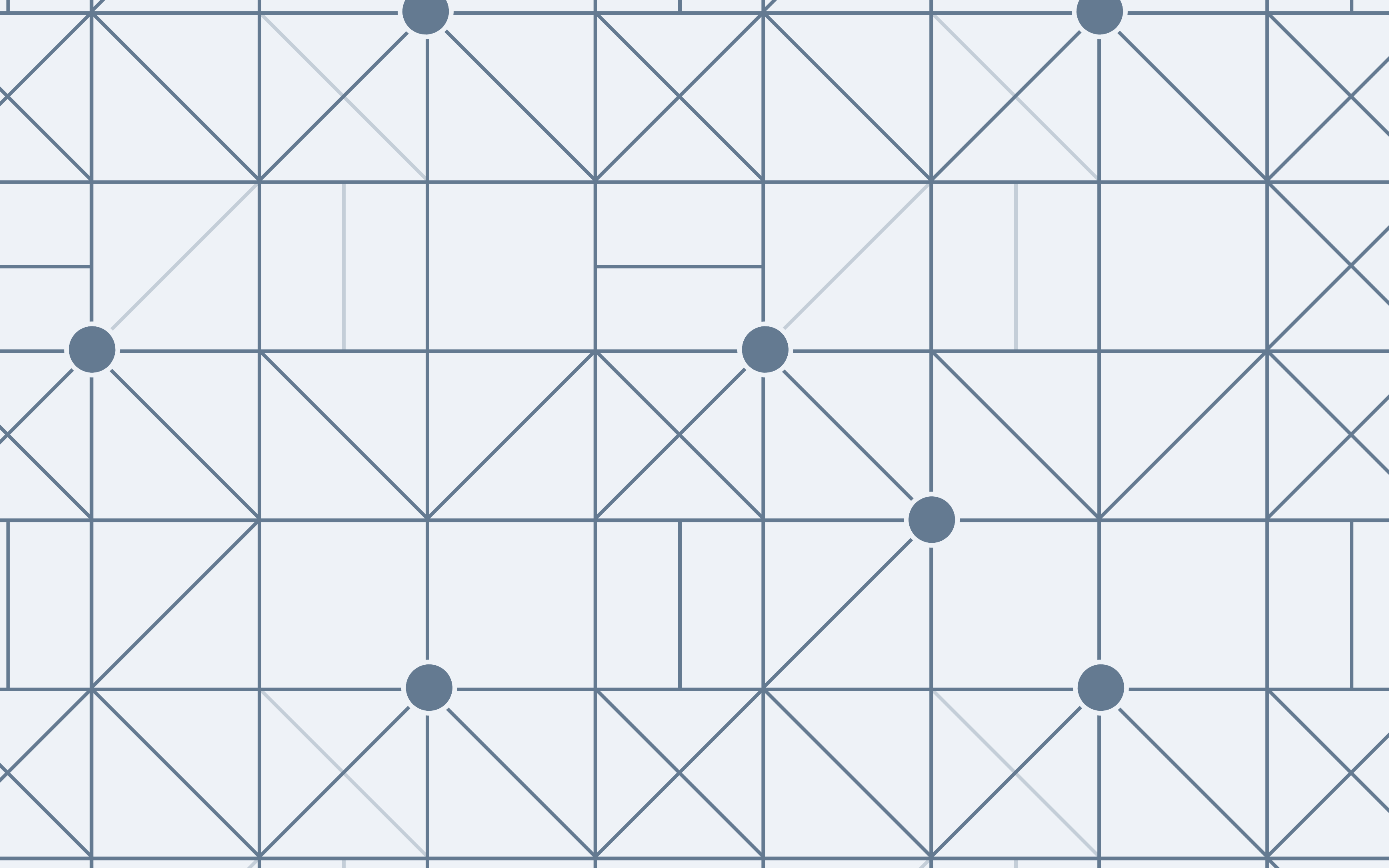


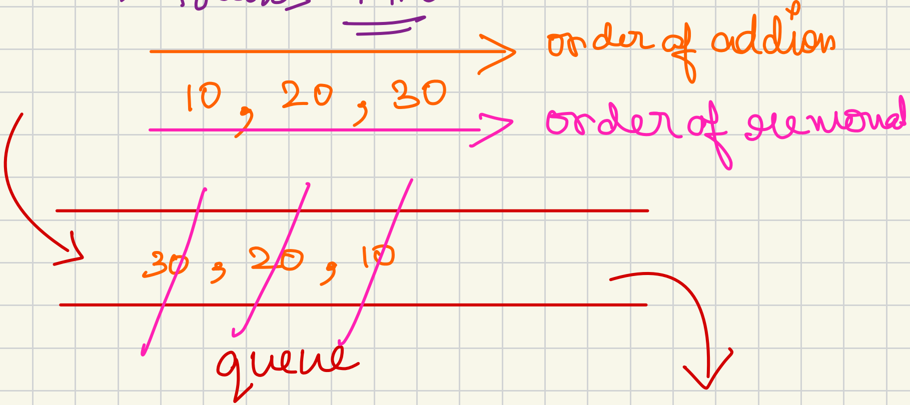
Queues.

Agenda.

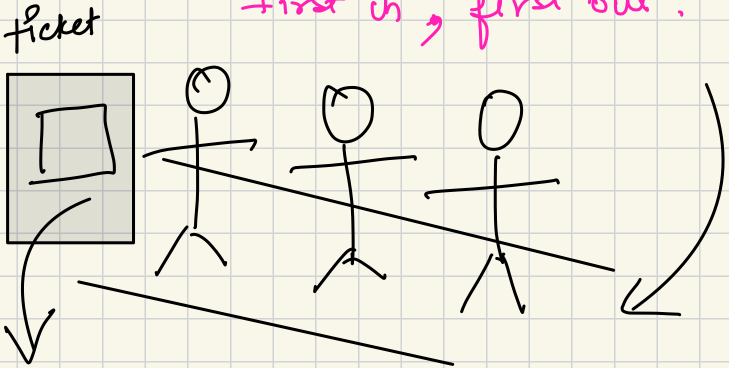
1. Introduction to queues
2. Basic ques on queues and stacks
3. 2 - stacks in a fixed Array
4. implementation of queues using 2 Stacks

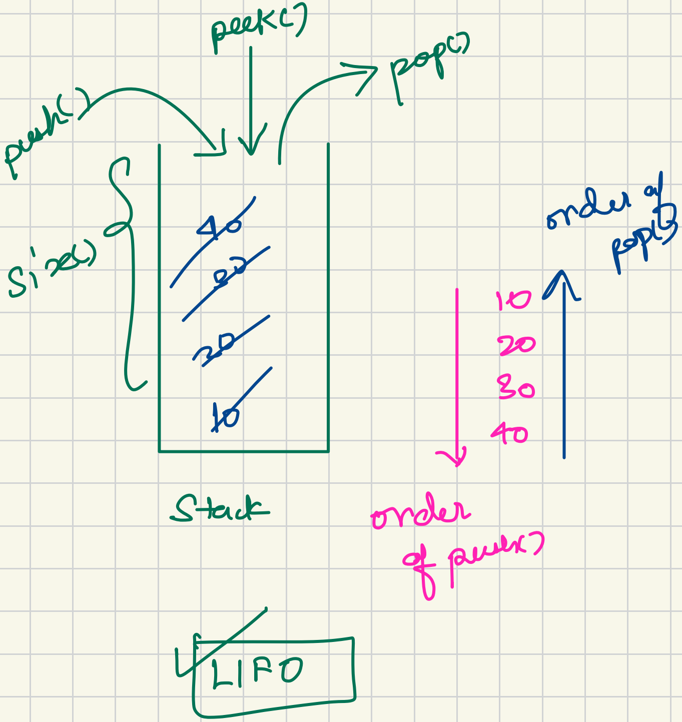
Queues .

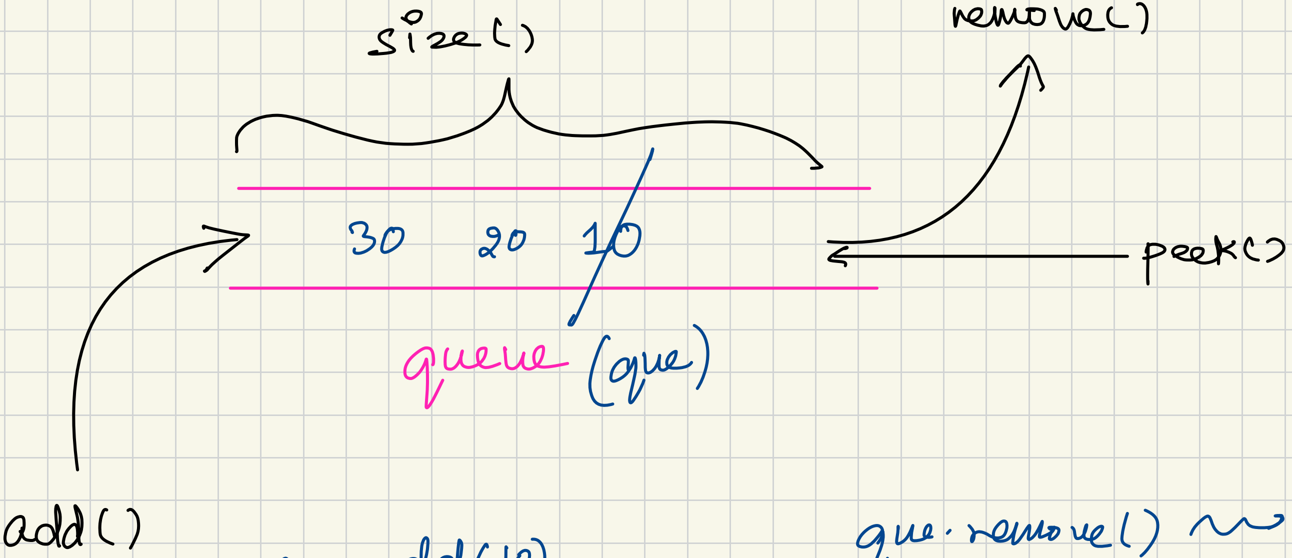
- ↳ linear Data Structures
- ↳ follows FIFO



first in , first out .







```
que.add(10)
que.add(20)
que.add(30)
que.size() ⇝ 3
que.peek() ⇝ 10
```

```
que.remove() ⇝ 10
que size() ⇝ 2
que.peek() ⇝ 20
```

Queue

→ interface in Java.

1. Queue<G> que_name = new ArrayDeque();
2. Queue<G> que_name = new LinkedList ();

Methods

1. add / offer
2. remove / poll
3. size
4. peek

} TC : O(1) , SC : O(1)

✓ **Enqueue** ∘

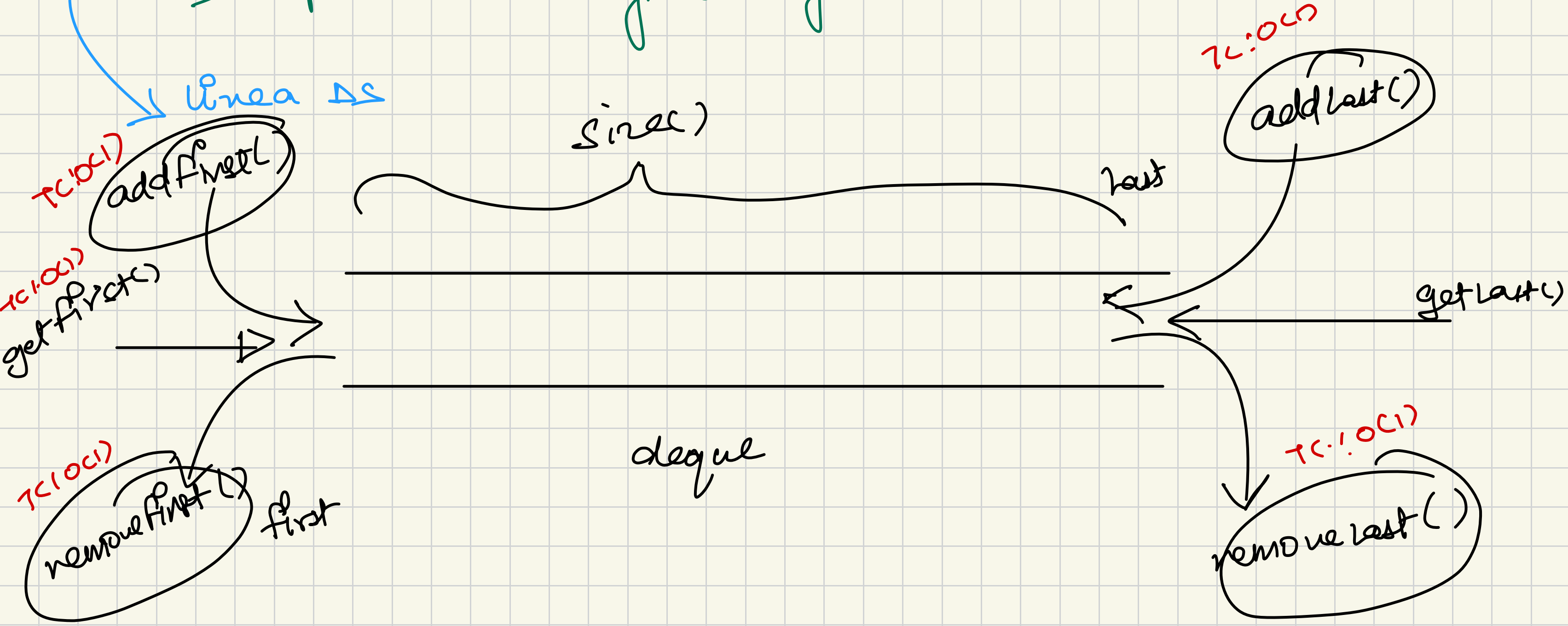
↳ {Enter + Queue} ↝ add/offer

Dequeue

↳ {Delete + Queue} ↝ remove/poll

Deque. {Doubly ended Queue}

→ implemented using doubly linked list



queue ?

↓

Linked List

```
class Node
{   int data;  → x
    Node next;  → y
}
```

✓ $\boxed{x+y}$

Deque

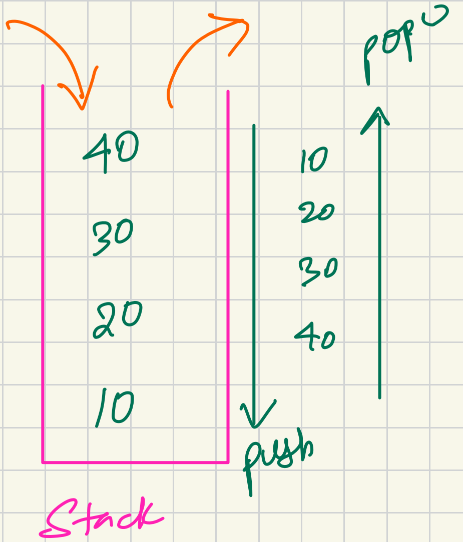
↓

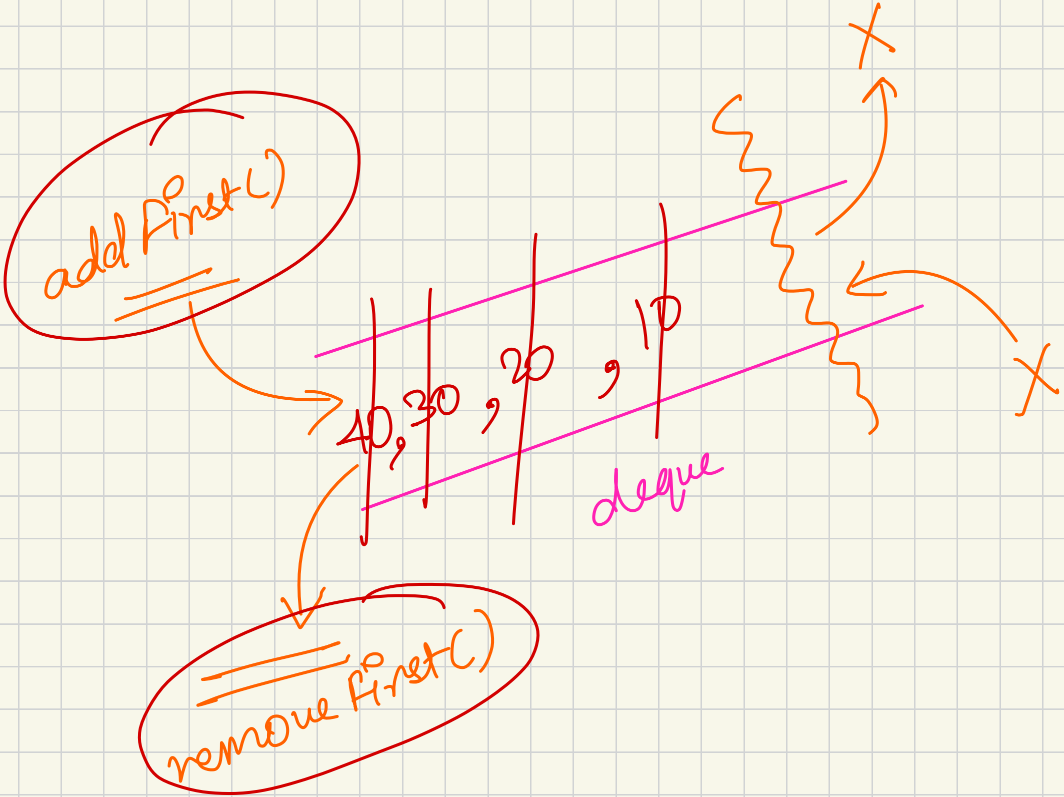
Doubly Linked List

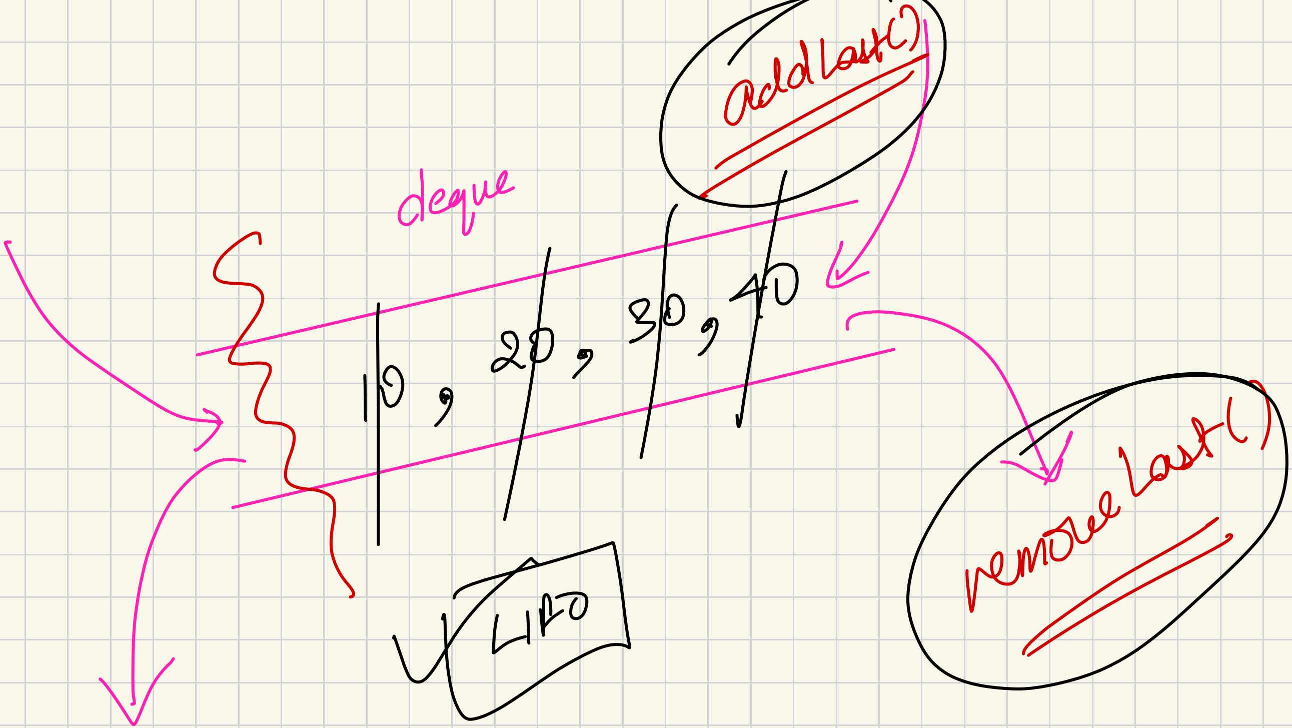
```
class Node
{   int data;  → x
    Node Next;  → y
    Node prev;  → y
}
```

✓ $\boxed{x+2y}$

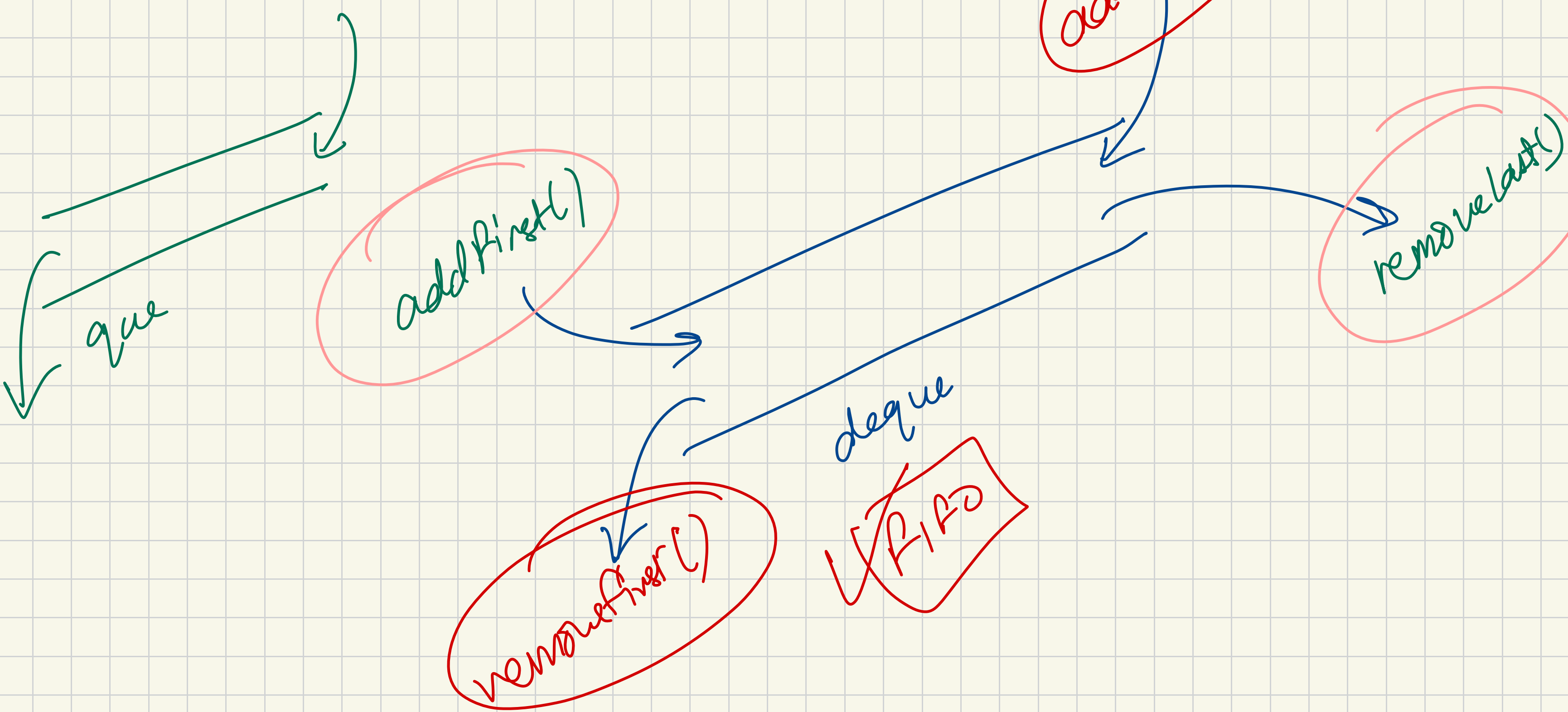
Q. Can you implement a Stack using Deque ?



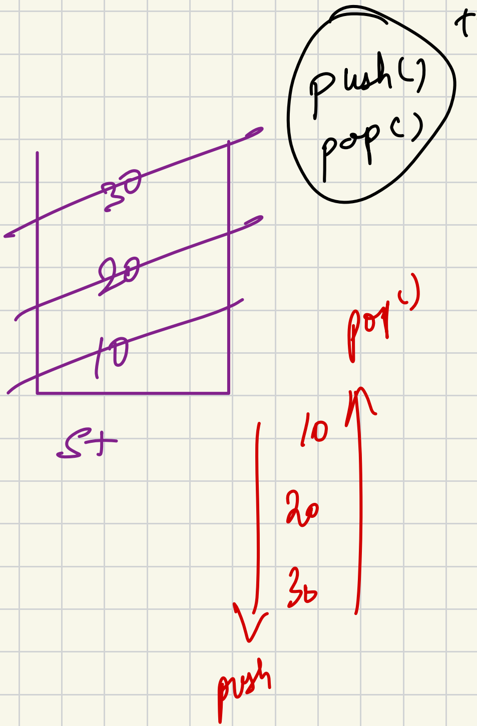



deque
10 , 20 , 30 , 40
addLast()
removeLast()
LIFO

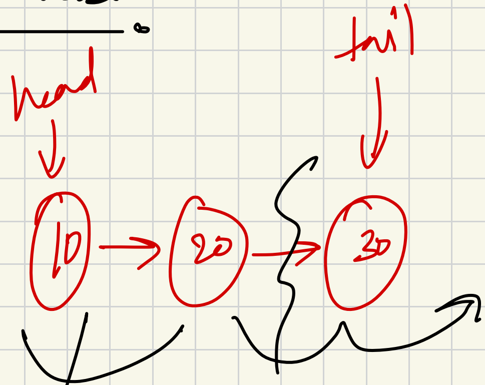
Q. Can you implement a queue using deque?



Design a stack using Linked list.



TC: O(1)

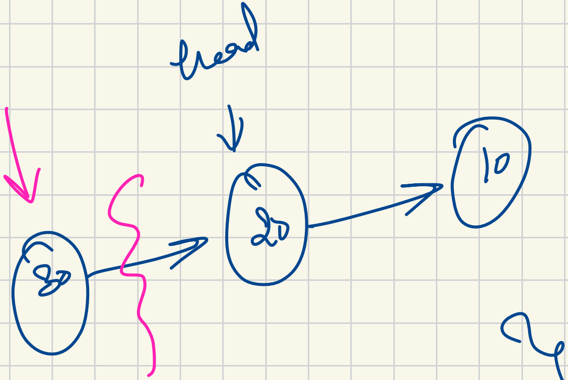


push() → O(1)

↳ addLast ll();

pop() → TC: O(N)

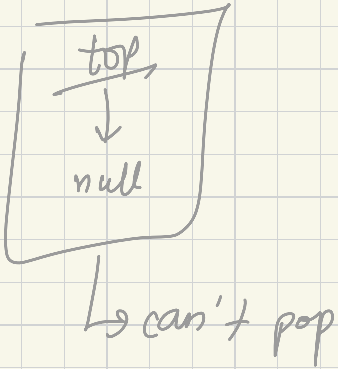
↳ removelast ll()



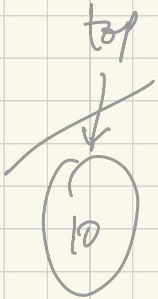
TC: O(1) push()
↳ addFirstLL() ✓

TC: O(1) pop()
↳ removeFirstLL() ✓

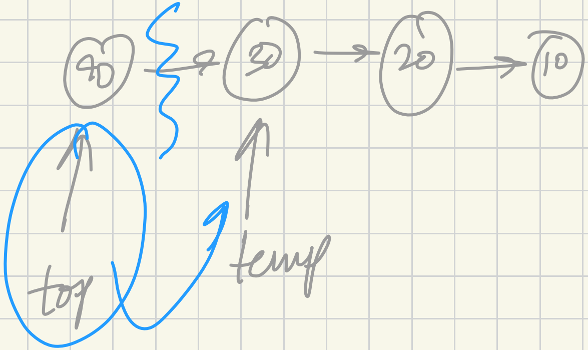
Case1

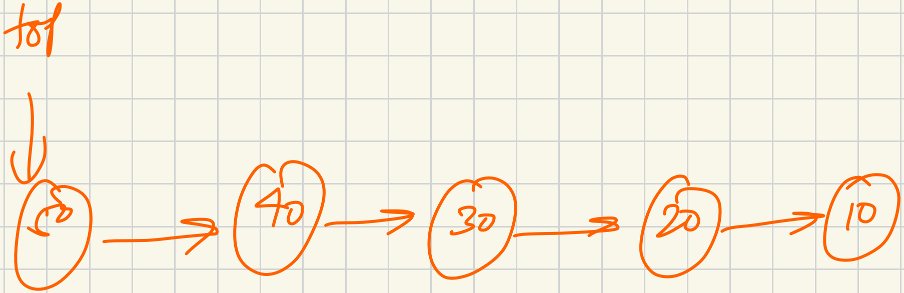


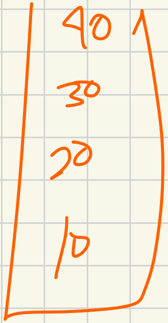
Case2



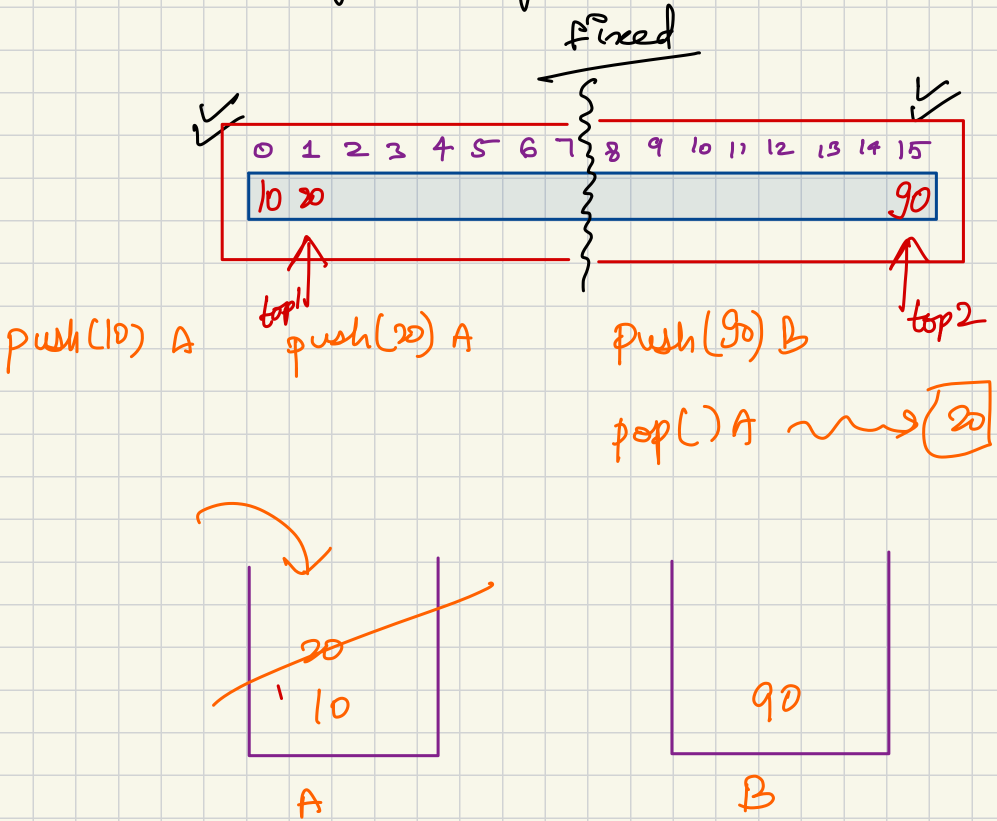
Case 3







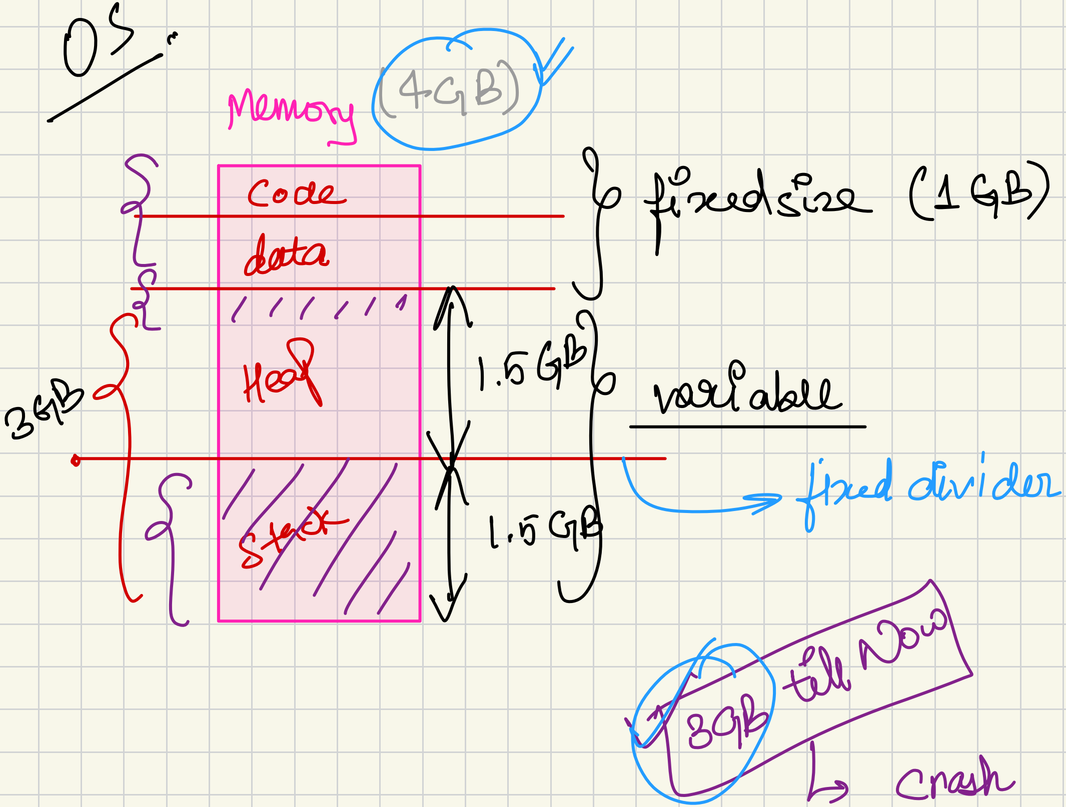
implement 2 stacks using an Array.



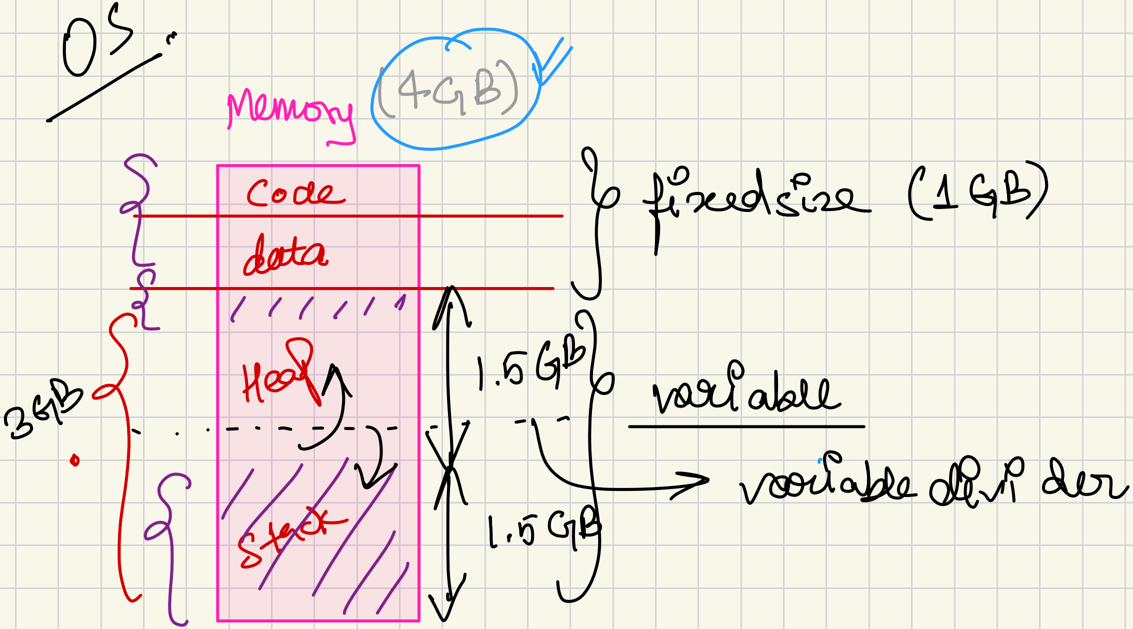
agree?

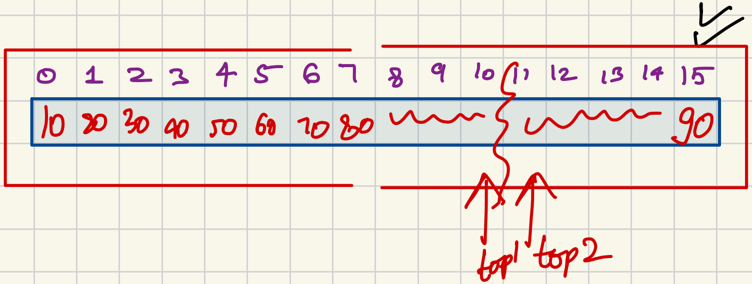
→ this will work fine }

OS :


Memory (4GB)
Code
data
Heap
Stack
fixedsize (1GB)
1.5GB
1.5GB
variable
3GB
fixed divider
3GB till Now
crash

OS :


Memory (4GB)
Code
data
Heap
Stack
fixedsize (1GB)
1.5GB
1.5GB
variable
variable divider
3GB



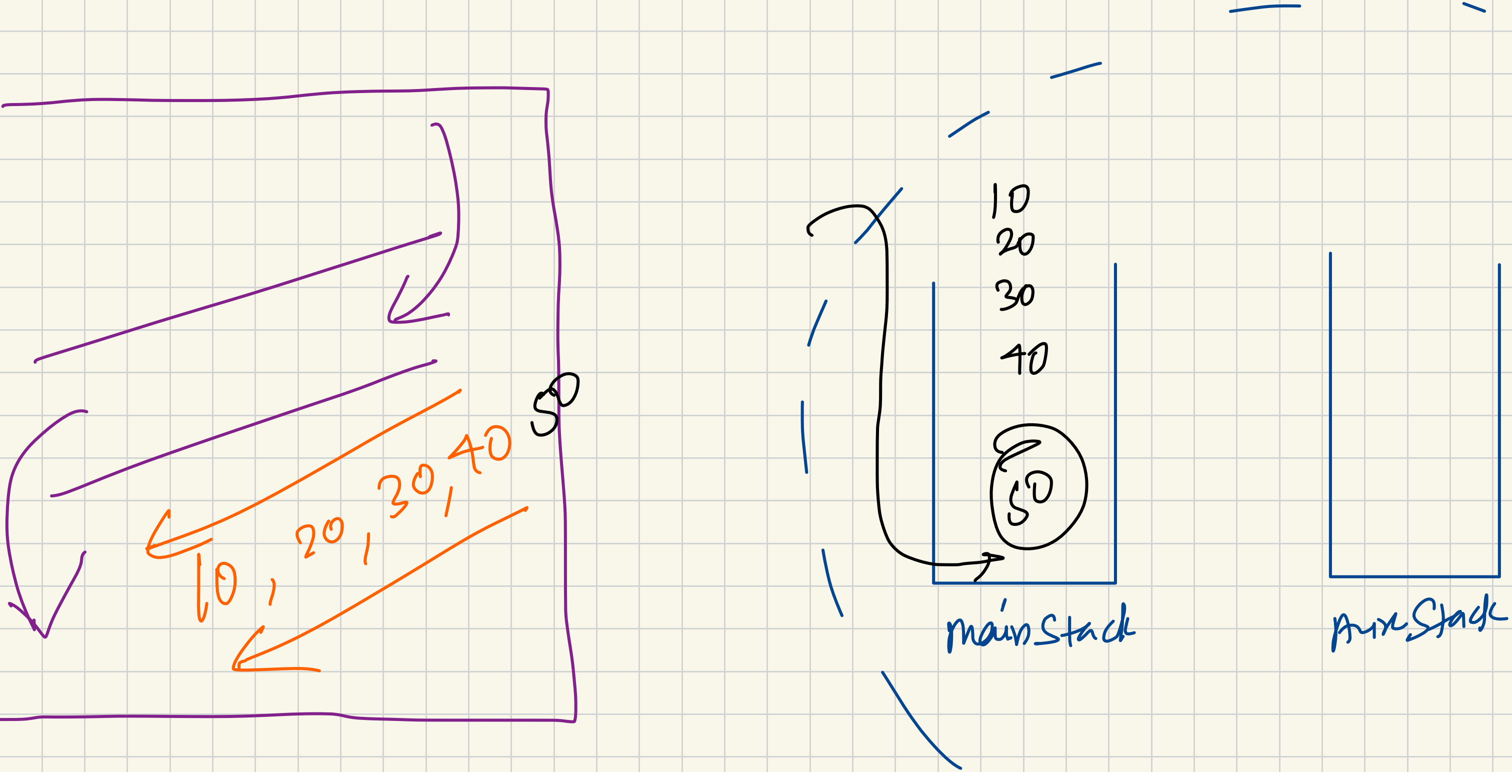
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	20	30	40	50	60	70	80								90

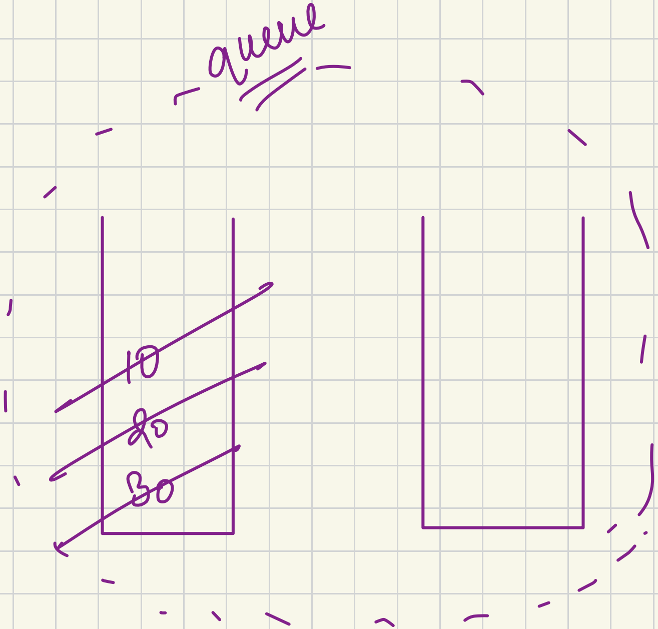
Implement Queue using 2 stacks

✓ Enqueue O(1)

✓ Dequeue O(1)

Implement queue using 2 stack where Dequeue TC: O(1)





add(10) add(20) add(30)

remove()