Queues.

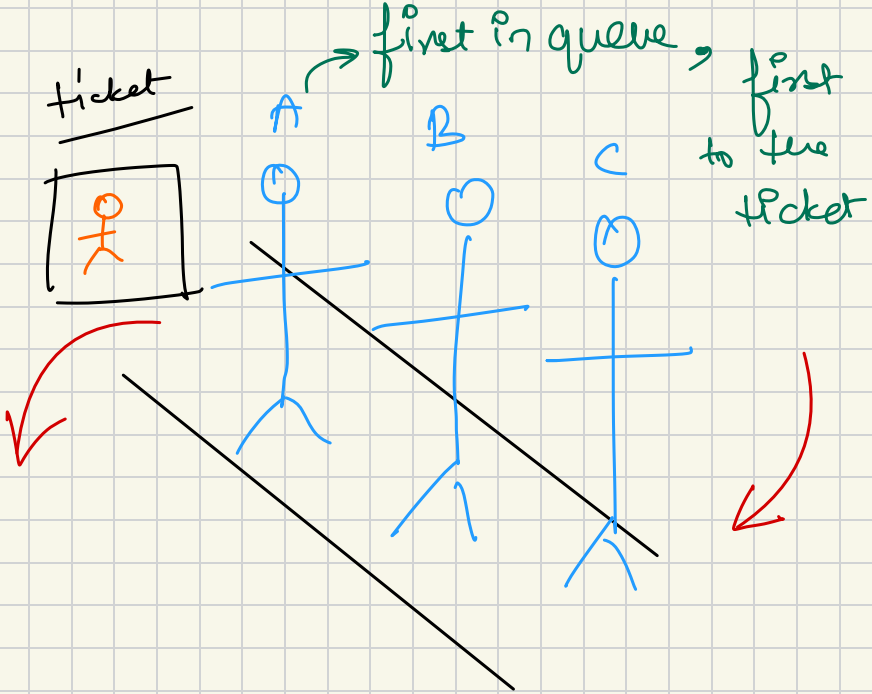
Agenda

- Introduction of queues
- Queue and Stacks easy problems
- 2 - Stacks in an array
- Implementation of queue using stack

Queues

↳ linear DS

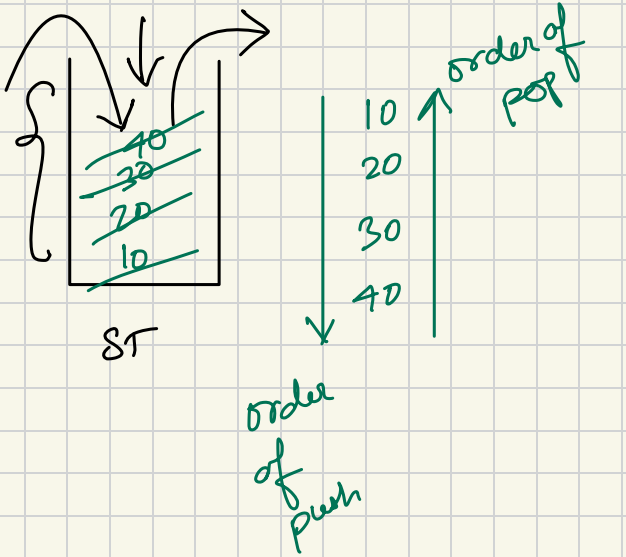
↳ follows FIFO { first in, first out }

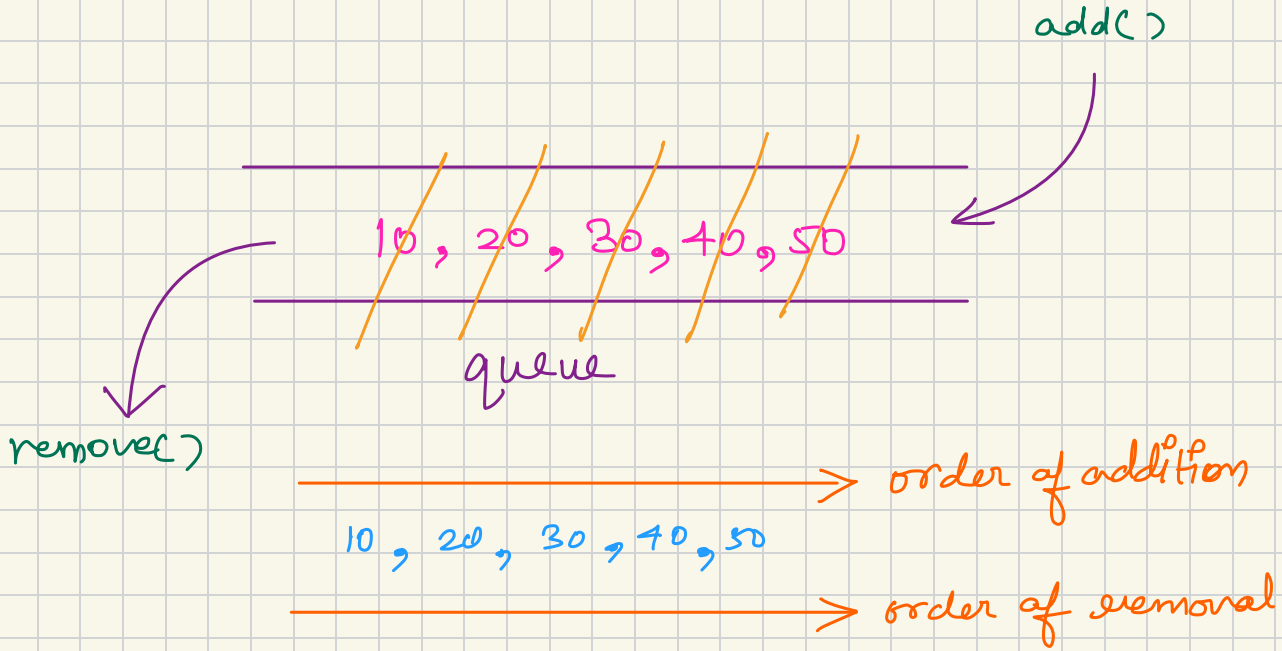


Stacks

↳ linear Data structure

↳ follows LIFO




add()
10, 20, 30, 40, 50
queue
remove()
order of addition
10, 20, 30, 40, 50
order of removal

Queue

↳ interface in Java

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

Methods

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

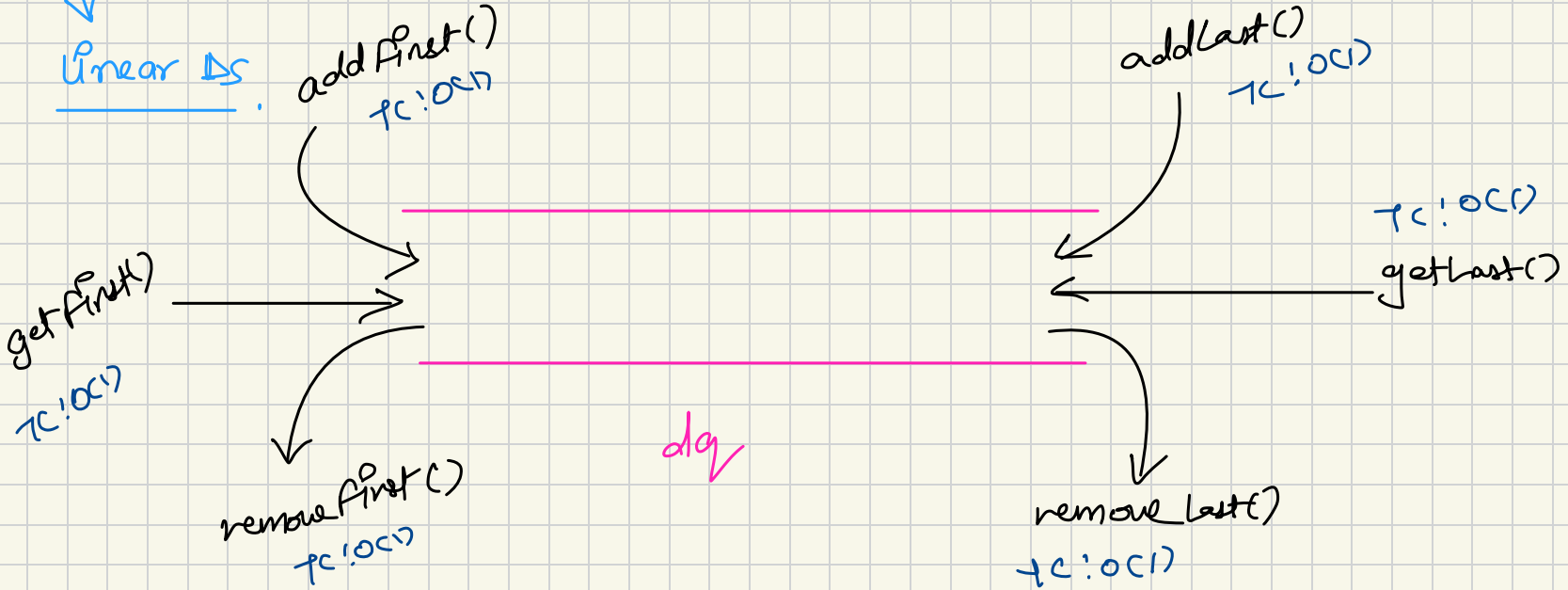
Enqueue ⟶ {Enter in Queue} add() / offer()

Dequeue ⟶ {Delete from Queue} remove() / poll()

Deque {Doubly Ended Queue}

→ implemented using doubly LinkedList.

Linear DS.




```
Deque<E> dq_name = new ArrayDeque<>();
```

Queue

↳ implemented using LL

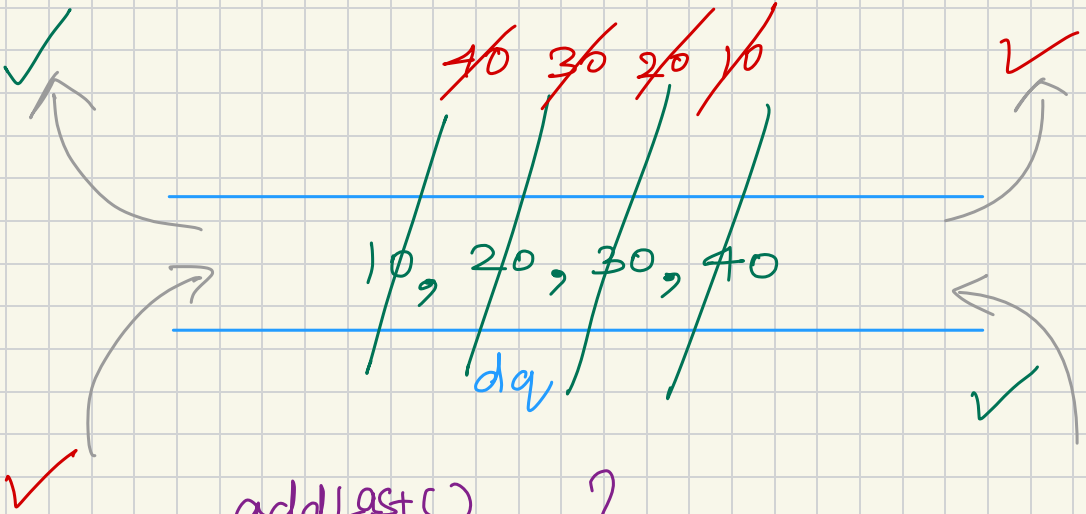
```
class Node
{
    Data;
    Node next;
}
```

Dequeue

↳ implemented using DLL

```
class Node
{
    Data;
    Node Next;
    Node Prev;
}
```

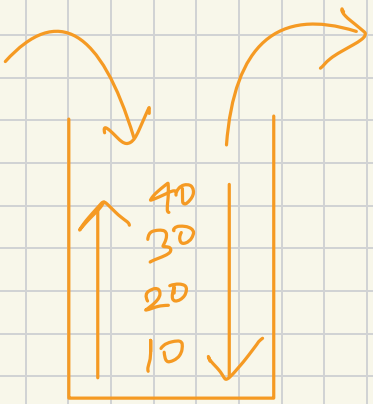
Q Can you implement a Queue using Deque

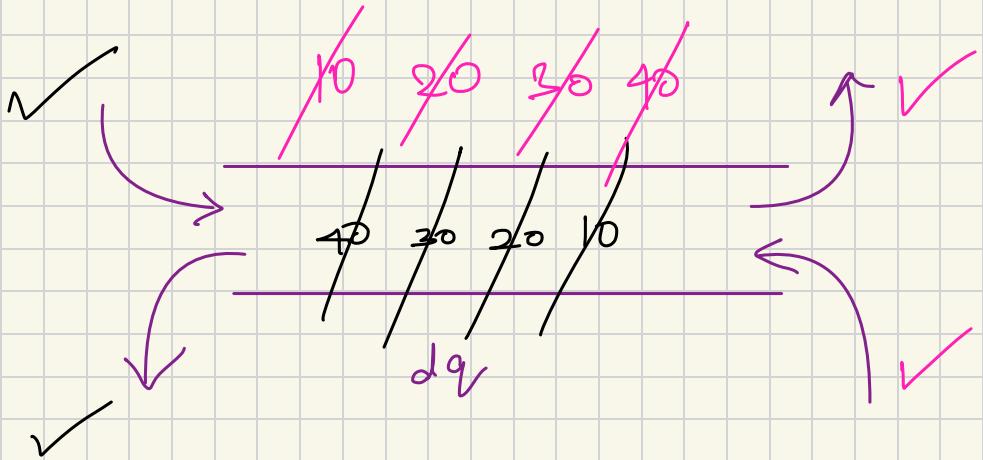


addLast()
removeFirst() } Queue using Deque.

addFirst()
removeLast() } Queue using Deque.

Q. Can you implement a stack using deque?



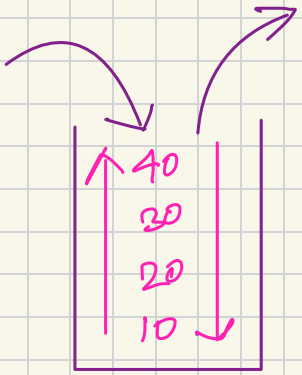


addFirst()
removeFirst()
} Stack using deque

addLast()
removeLast()
} Stack using deque

Design a Stack using Linked list.

using Linked list.

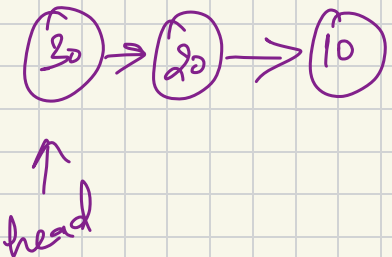


push(10)
push(20)
push(30)
push(40)
peek()
remove() → 40

use

push() : ~~addLast()~~ LL TC:O(1)
pop() : ~~removeLast()~~ LL : TC:O(N)

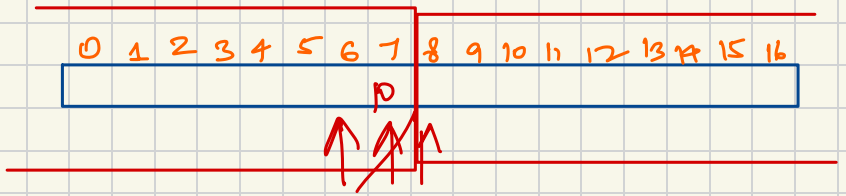
Bad TC



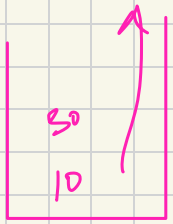
push() : addFirst()
pop() : removeFirst()

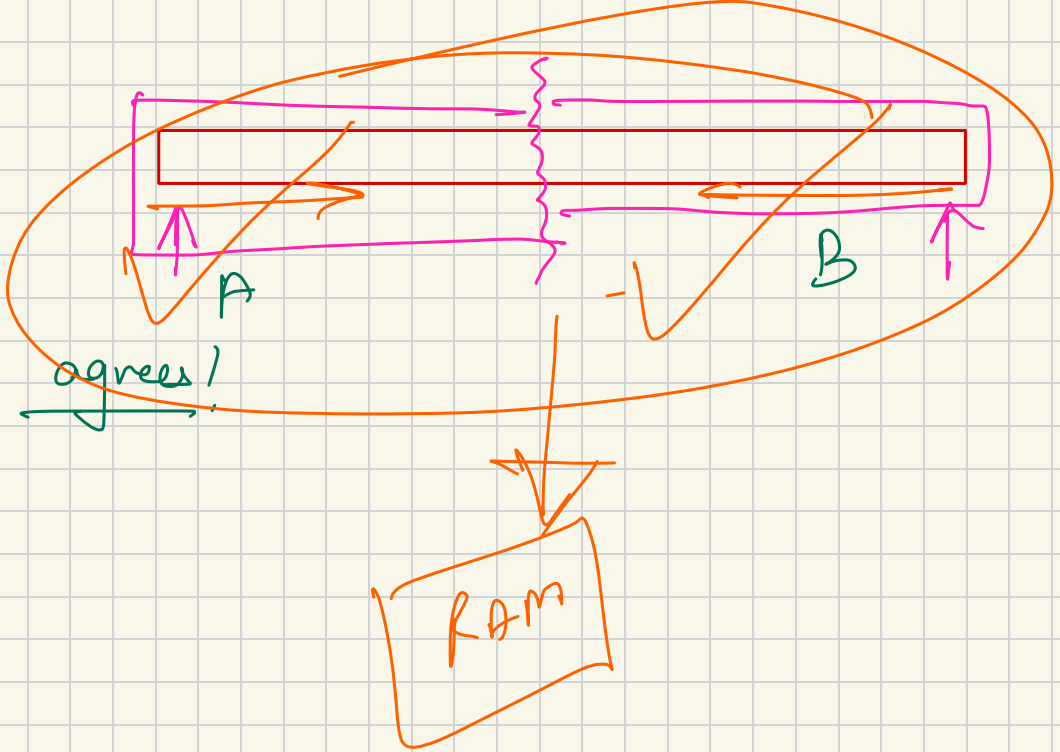
TC:O(1) ✓

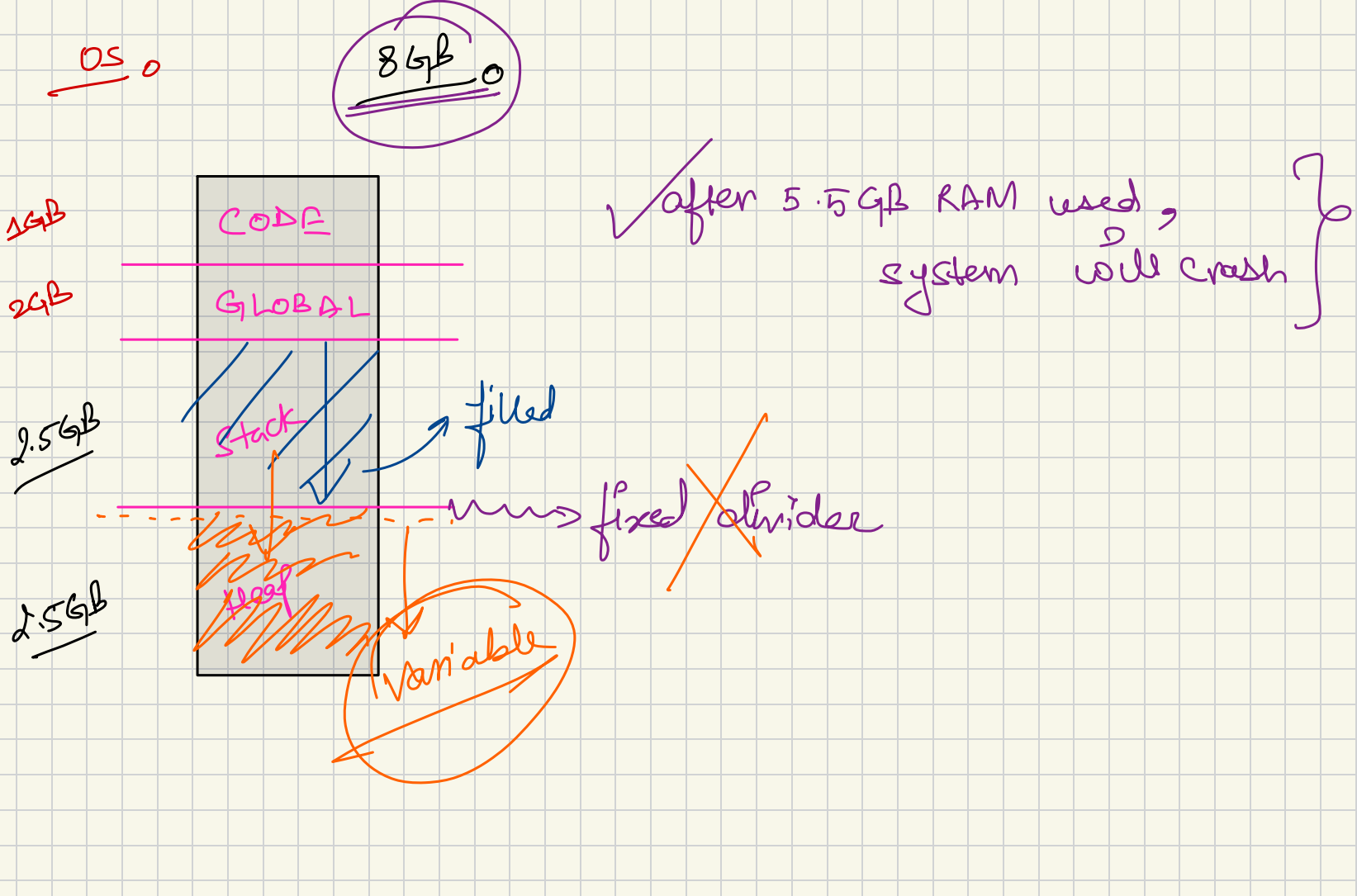
Implement 2 Stacks using an Array

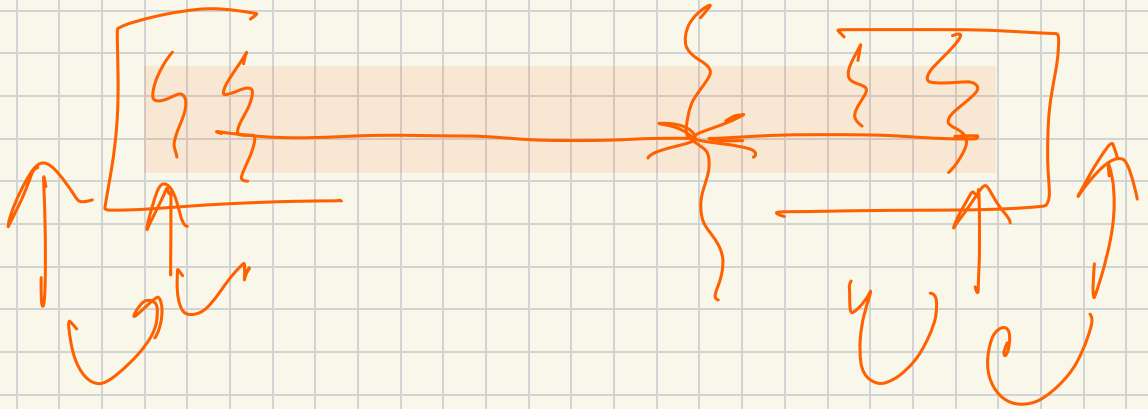


push(10), push(20), push()

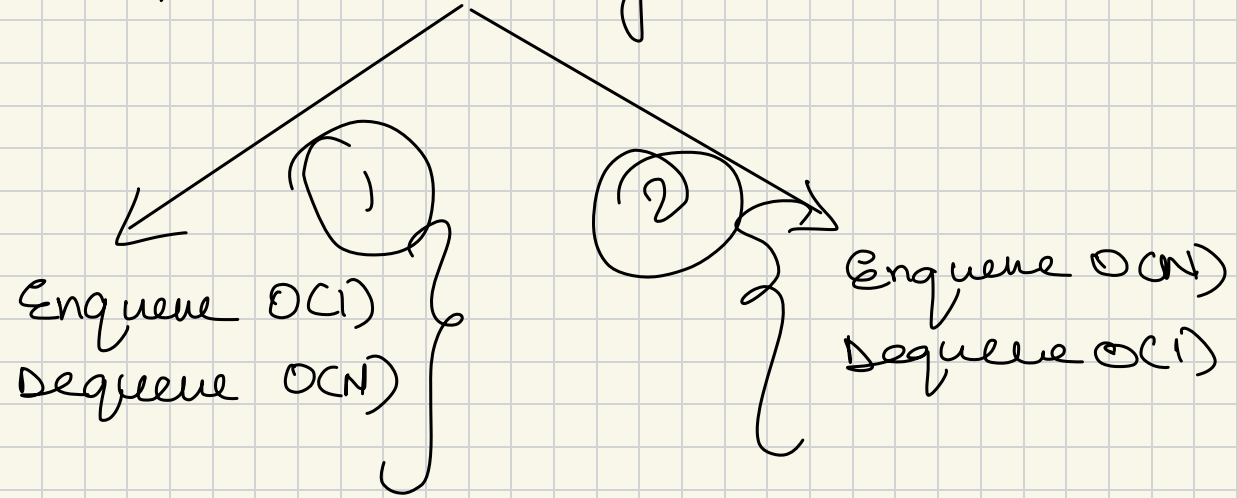



A
B
agrees!
RAM

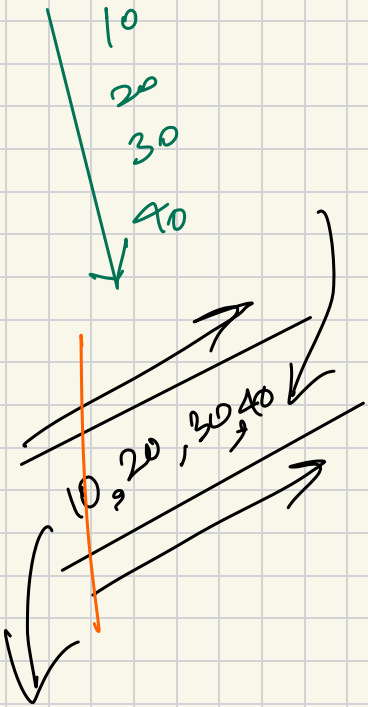

OS
8GB
1GB
CODE
2GB
GLOBAL
2.5GB
Stack
filled
fixed divider
Heap
2.5GB
Variable
after 5.5GB RAM used, system will crash



Implement Queue using 2-Stacks

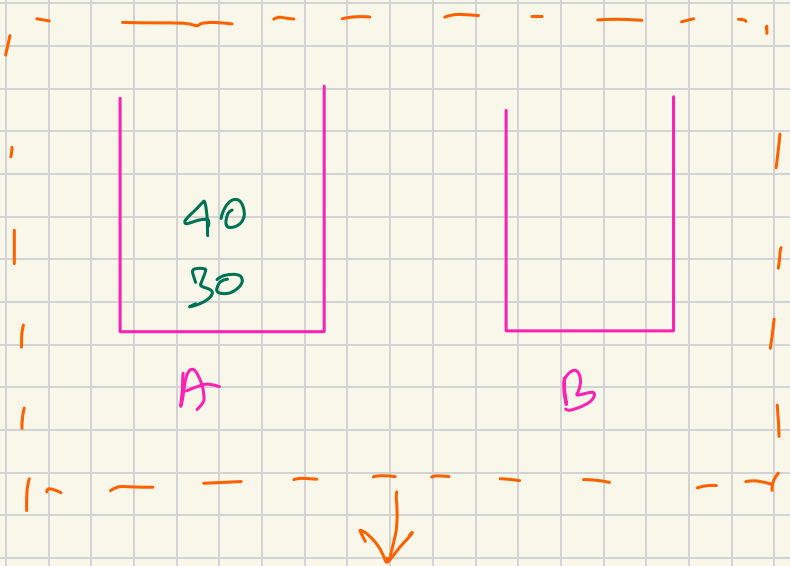


Implement Queue using 2 Stacks, where enqueue O(1)



10 20

Dequeue TC: O(N)



Queue