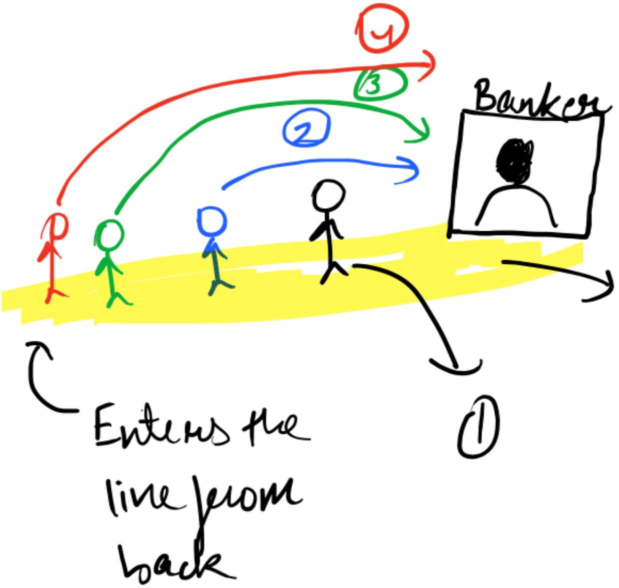
Queue

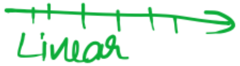
Monday, 29 July 2024 8:56 PM

Agenda
- → Introduction to Queue ✓
- → Easy Problem ✓
- → 2 Stacks in 1 Array ✓✗
- → Implementation of Queue ✓
  - → using LL
  - → using 2 Stacks

Solve I/W for Queue



Leaves the line

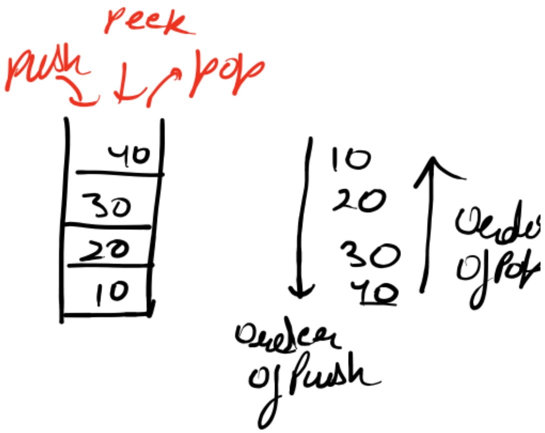


First in Queue, First to leave the Queue

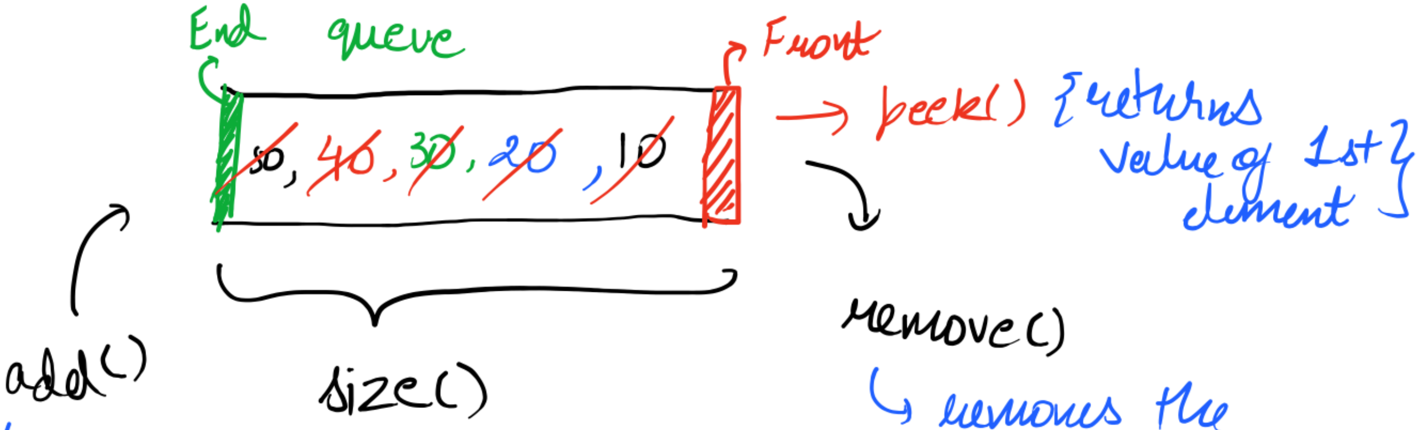
⇓

FIFO
↳ First In First Out

Linear DS + LIFO



FIFO !!



add()
↳ adds element at the end of the queue

remove()
↳ removes the element from front of the queue.

order of adding →

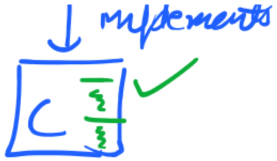
10, 20, 30, 40, 50

order of removal

FIFO

Queue
↳ interface in Java

① Queue<E> quename = new ArrayDeque<>();



② Queue<E> quevename = new LinkedList <>();

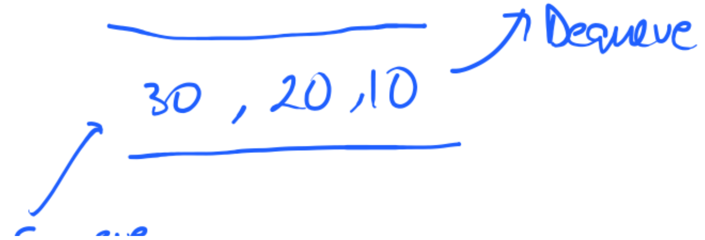
Methods:

Enqueue ← ① add ( ) / offer()
Dequeue ← ② remove() / poll()
③ peek()
④ size

For the operations
TC: O(1)
SC: O(1)

Method

* Enqueue → { Enter in queue } add() / offer()
* Dequeue → { Delete from queue } remove() / poll()

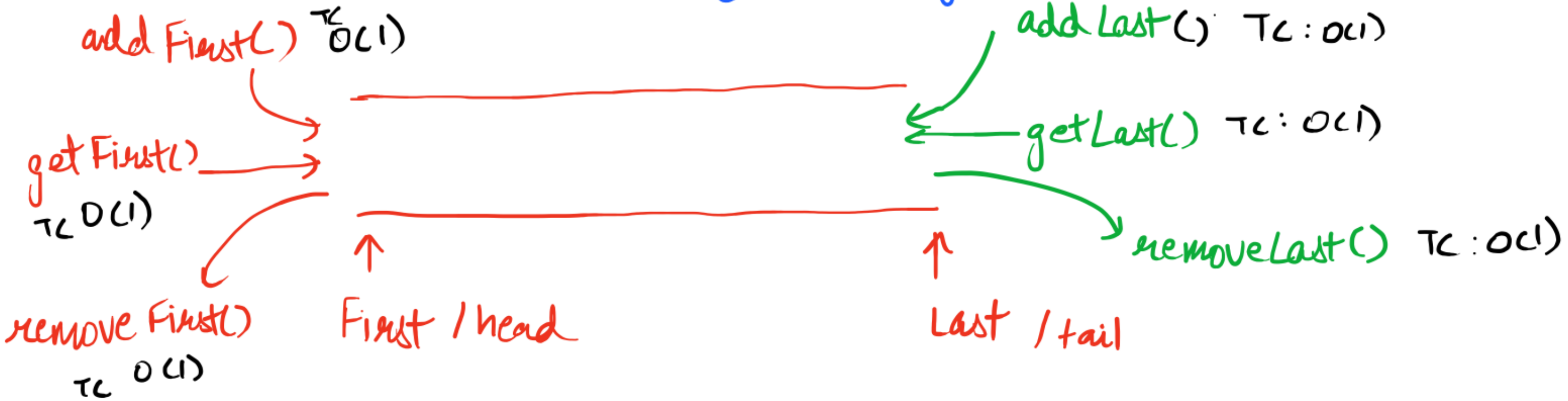


Enqueue

Deque { Double Ended Queue }

Linear DS

Implemented using Doubly Linked List



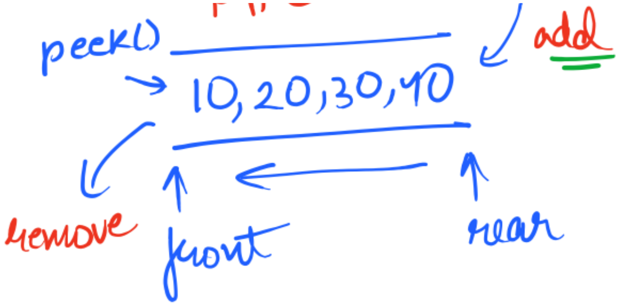
⇒ Deque <E> deque = new ArrayDeque<>();

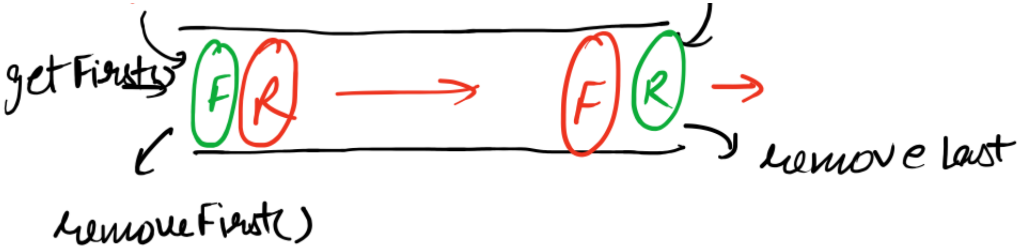
Q1) Implement Queue using Deque?

addFirst()

add Last()

FIFO

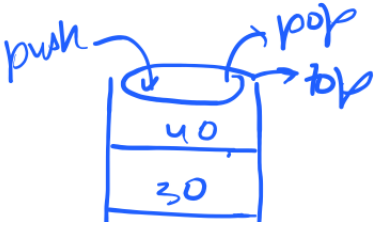


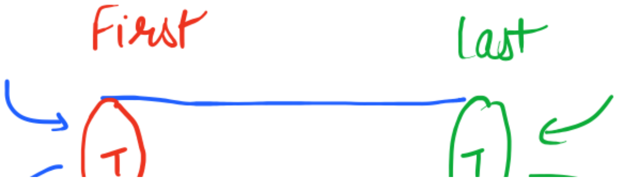


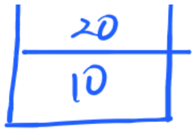
remove()	removeFirst()
add()	addLast()
peek()	getFirst()

add	addFirst()
remove	removeLast()
peek	getLast()

Q2) Implement a stack using a deque?







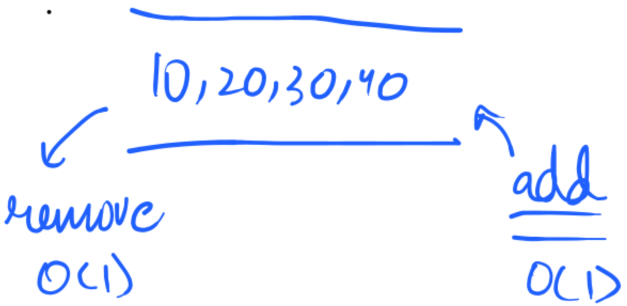
pop() { removeFirst()
push() { addFirst()

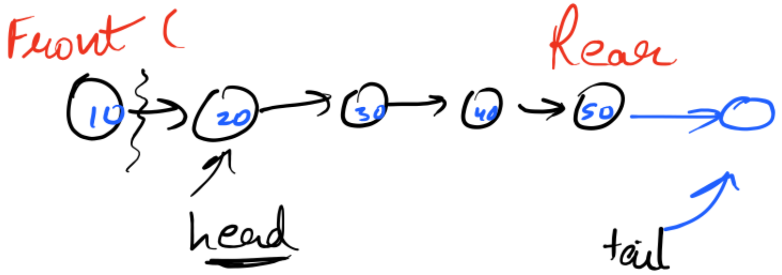
LIFO? ✓

push { addLast()
pop { removeLast()

LIFO

Design a Queue using LL.

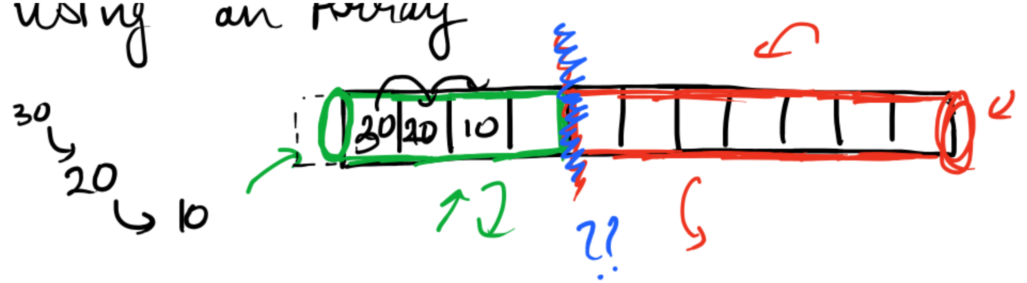




add → addTail() O(1)
remove → removehead() O(1)
peek() → getHead()

2. Stack using Array

2 Stack using an Array

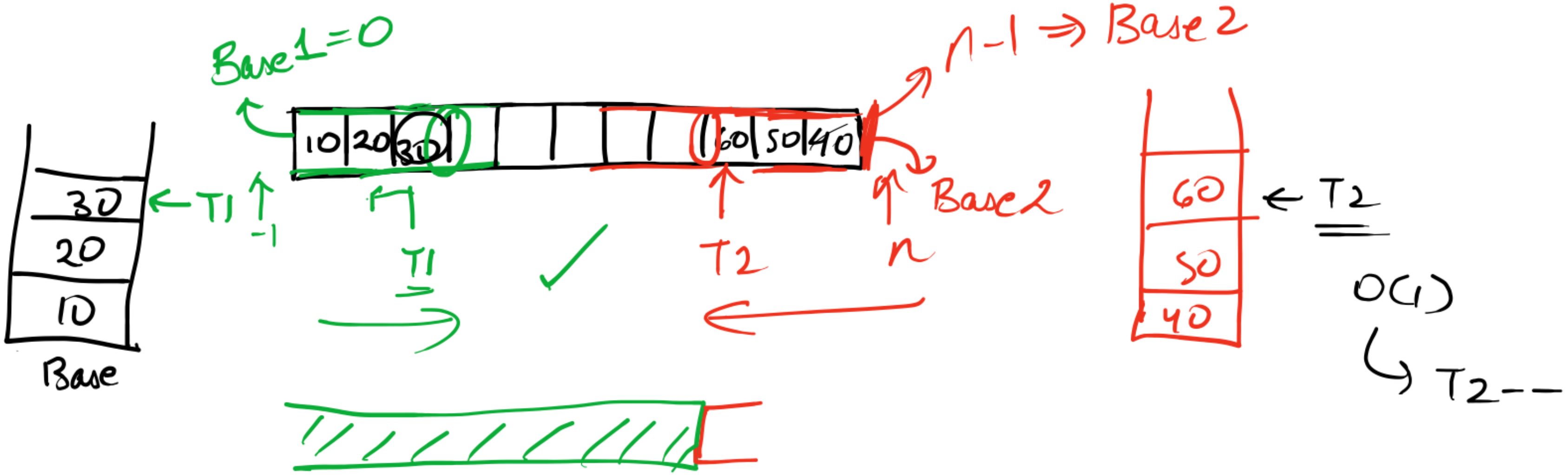


approach?

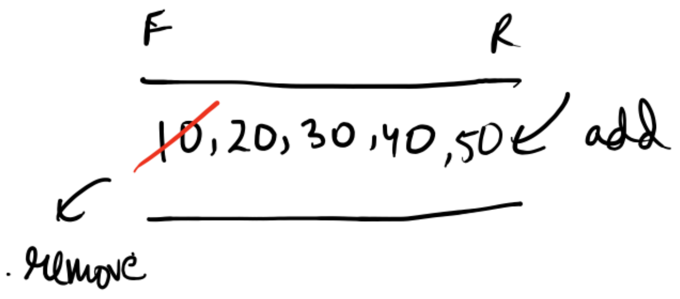
Stack
→ LIFO
↙
all operations O(1)
from one end only

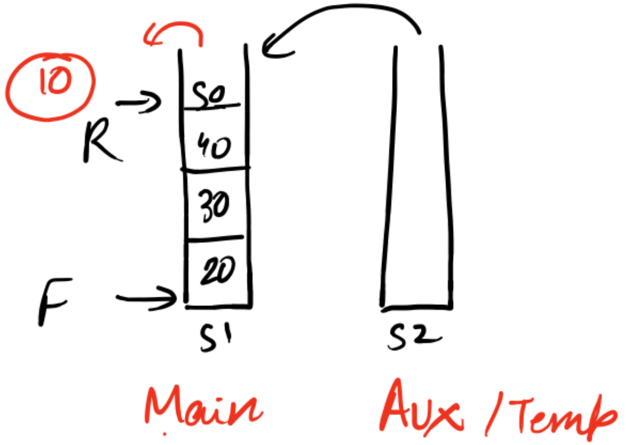
if 0 → top1
n-1 → top2 ⇒ How to identify base of stack?

↳ shifting would be required?
O(n) ✗



Implement a queue using 2 Stack





Enqueue / Add → Push O(1)

Dequeue / Remove → We need to remove element from base of S1
O(N)

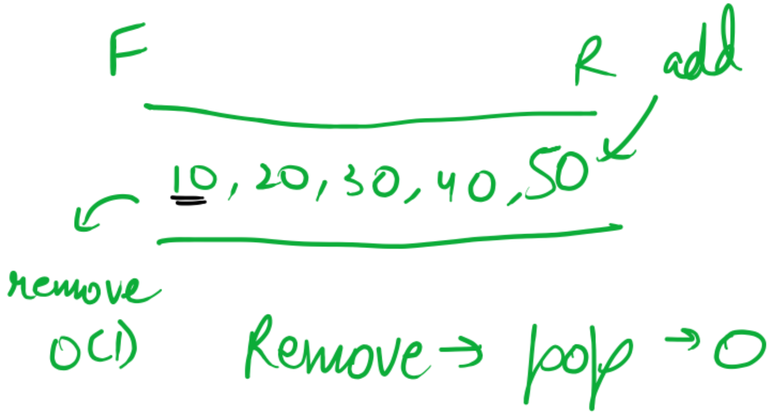
Steps
↳ Shift N-1 elements from S1 to S2
→ Remove base element from S1
↳ Move all elements back to S1.

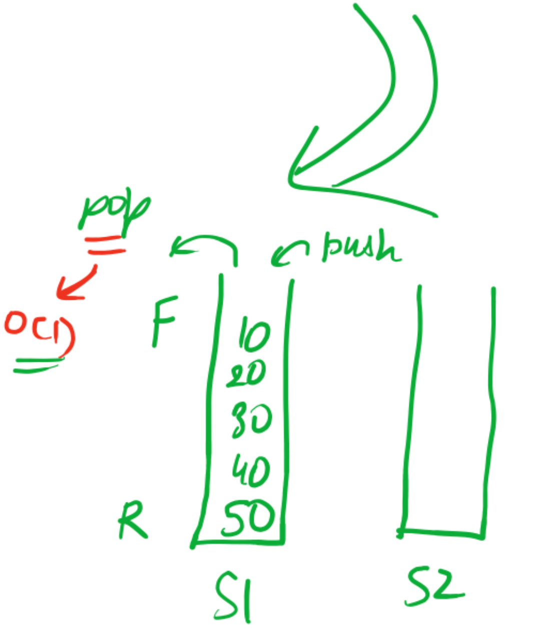
Queue Using 2 Stacks

Queue using 2 stacks

Enqueue O(1)
→ more preference to add in O(1)
→ Dequeue → O(N)

Dequeue O(1)
→ Remove will work in O(1)
→ Enqueue → O(N)





Remove → pop → O(1)

Add in reverse order

⇓ Steps ??

1) Move all elements to S2
2) Add x in S1
3) Move all elements from S2 to S1.

Created with OneNote.