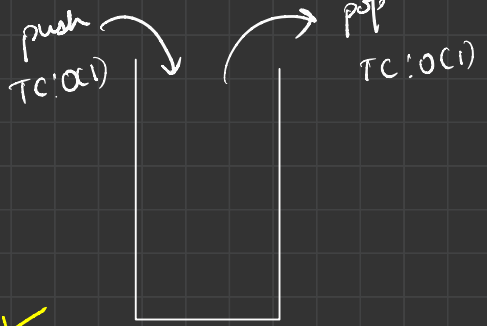
Queues.

Stacks

LIFO

↓

{ last in, first out }



push(40)

push(30)

push(50)

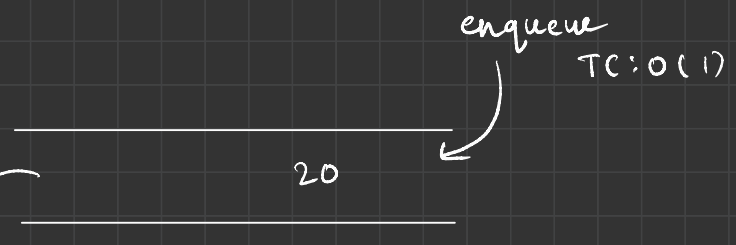
pop() → 50

pop() → 30

pop() → 40

Stacks.

Queue



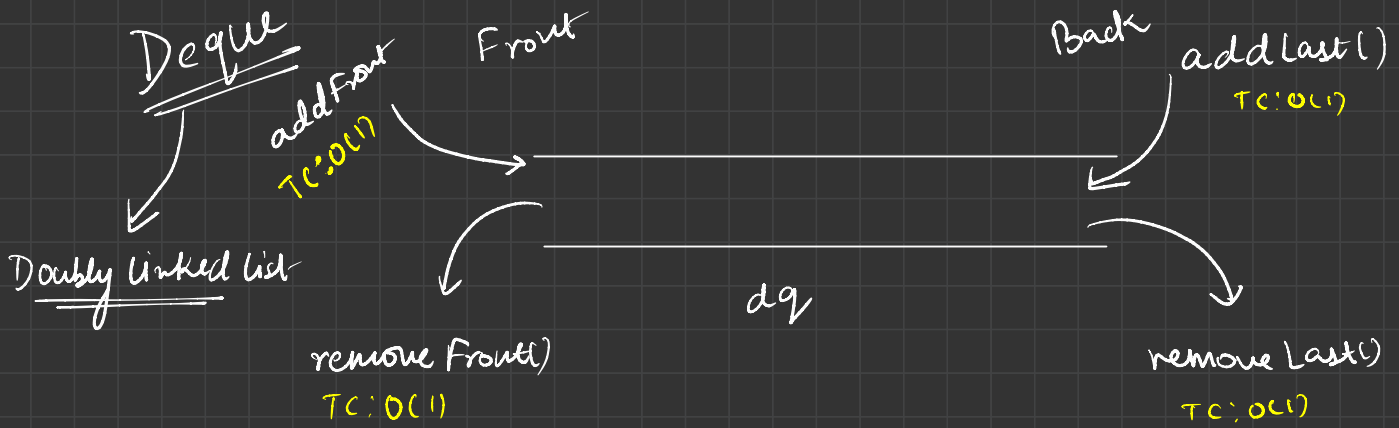
add (30)
add(40)
add(20)

dequeue
TC: O(1)

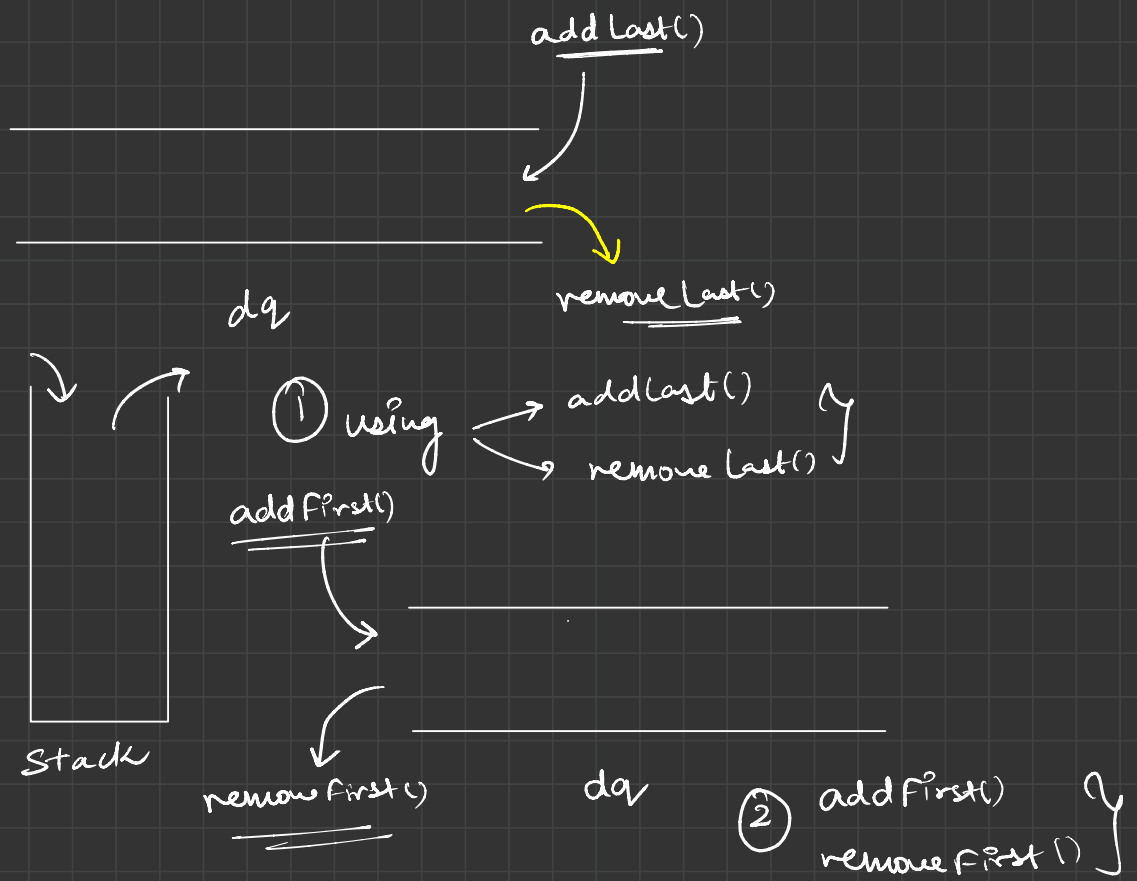
Queue
(FIFO)

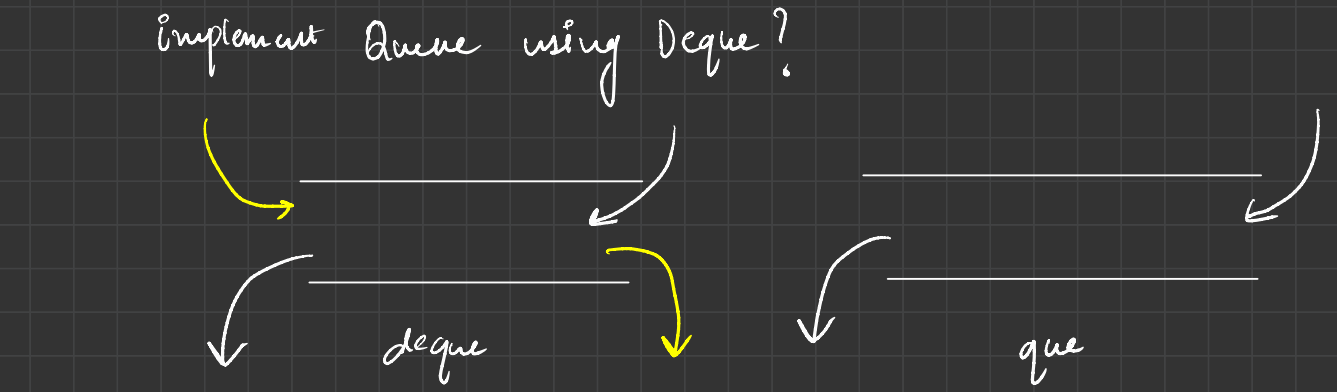
↳ first in first out

remove() → 30
remove() → 40


Deque
addFront
TC: O(1)
Front
Back
addLast()
TC: O(1)
Doubly Linked List
dq
removeFront()
TC: O(1)
removeLast()
TC: O(1)

Can you guys implement a stack using a deque?





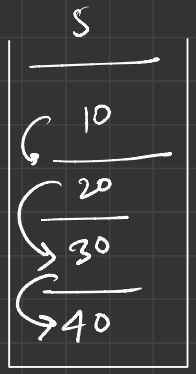
① { ① addLast() TC: O(1)
② removeFirst() TC: O(1)

② { ① addFirst()
② removeLast()

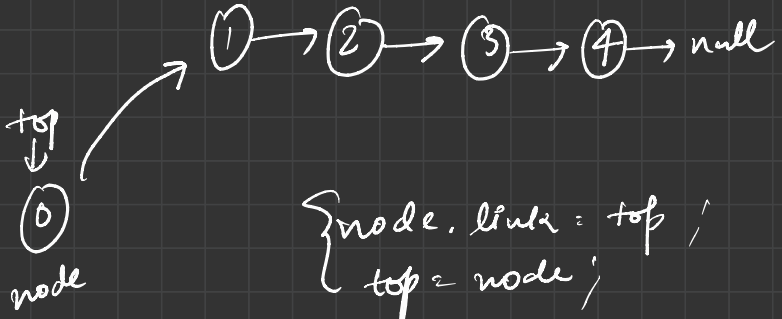
Q Stack using a linked list

push() → O(1)

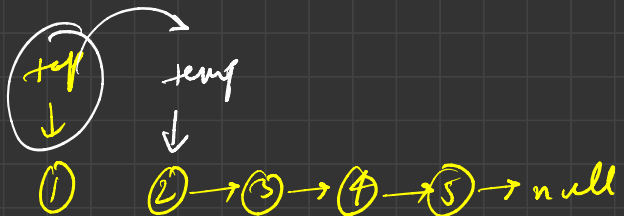
pop() → O(1)



{ addFirstLL()
removeFirstLL()

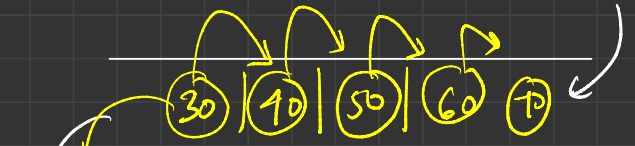



```
{node.link = top;
 top = node;
```



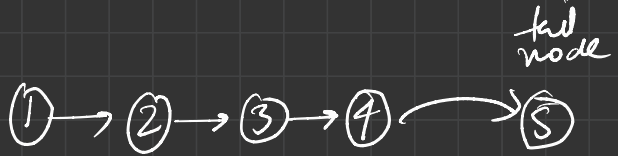

```
{ Node temp = top.link;
  top.link = null;
  top = temp;
```

Queue using LinkedList



Enqueue → addLast() O(1)

dequeue → removeFirst() O(1)

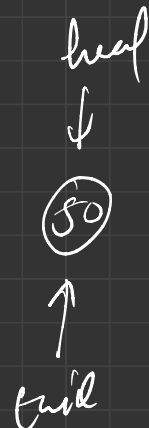


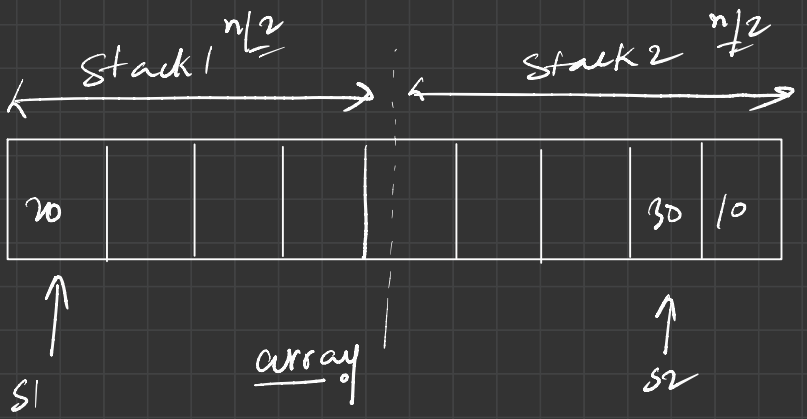

```
{ tail.next = node;
  tail = node;
```

size = 0

head → null

tail → null





Implement 2 stacks using an array.

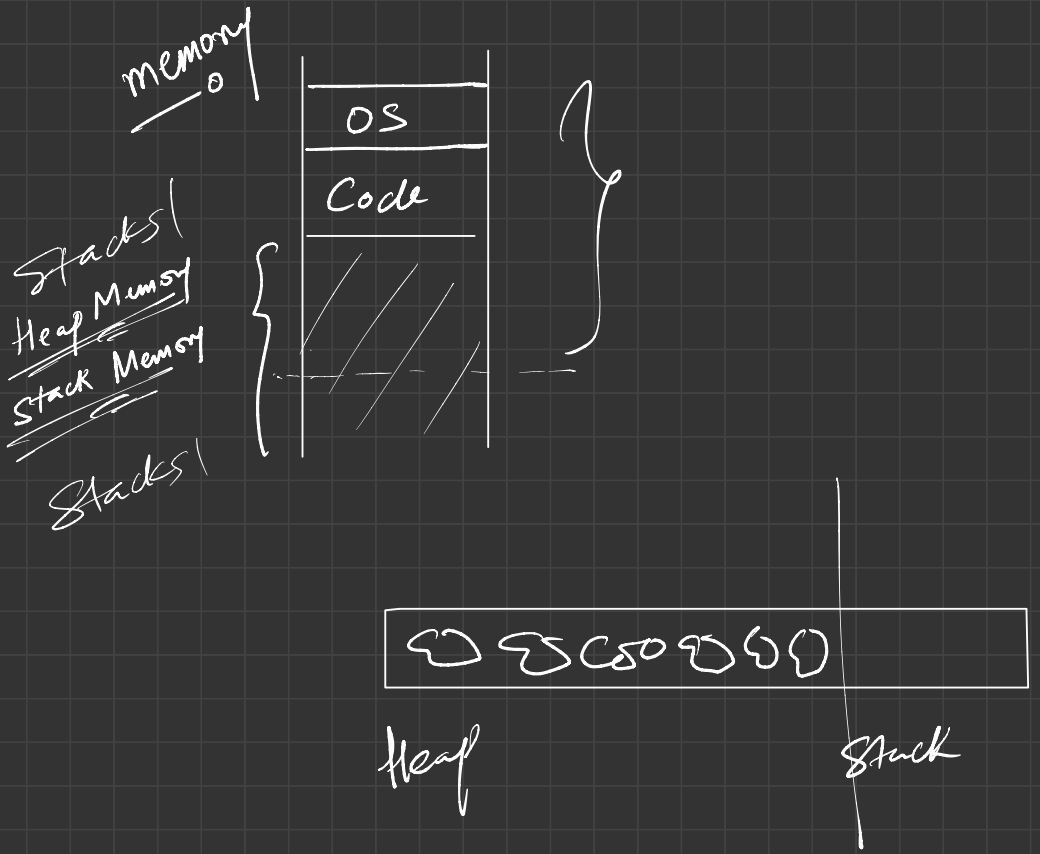
S1. push(20)

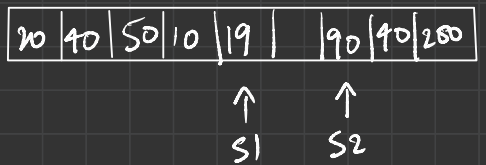
S1. push(30)

S1. pop()

S2. push(10)

S2. push(30)


Memory
0
OS
Code
Stacks
Heap Memory
Stack Memory
Stacks
Heap
Stack



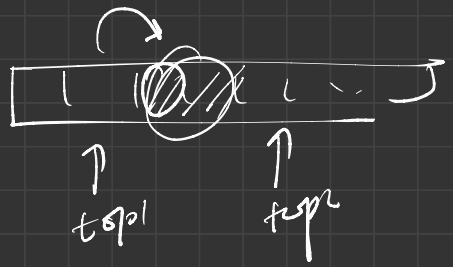
S1

push(20)
push(40)
push(50)
push(10)
push(19)
push(30)
pop()

S2

push(200)
push(40)
push(90)
push(70)

↳ Stack Overflow!



$(top1 + 1) < top2$

```
// Method to push an element x to stack1
void push1(int x)
{
    // Your code here
    if (top1 + 1 < top2) {
        top1 += 1;
        arr[top1] = x;
    }
}

// Method to push an element
// x to stack2
void push2(int x)
{
   // Your code here
    if (top2 - 1 > top1) {
        top2 -= 1;
        arr[top2] = x;
    }
}
```

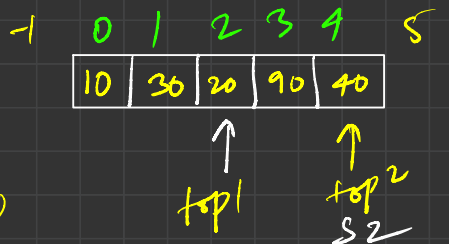
```
// Method to pop an element from first stack
void pop1()
{
    // Your code here
    if (top1 == -1) {
        System.out.println(-1);
    } else {
        System.out.println(arr[top1]);
        top1 -= 1;
    }
}

// Method to pop an element
// from second stack
void pop2()
{
    // Your code here
    if (top2 == size) {
        System.out.println(-1);
    } else {
        System.out.println(arr[top2]);
        top2 += 1;
    }
}
```

S-1 4 72

0/1

90
100
-1



S1 pop()

push(10) push(20)
push(30) push(90)

S2

push(100) pop()
pop() push(40)

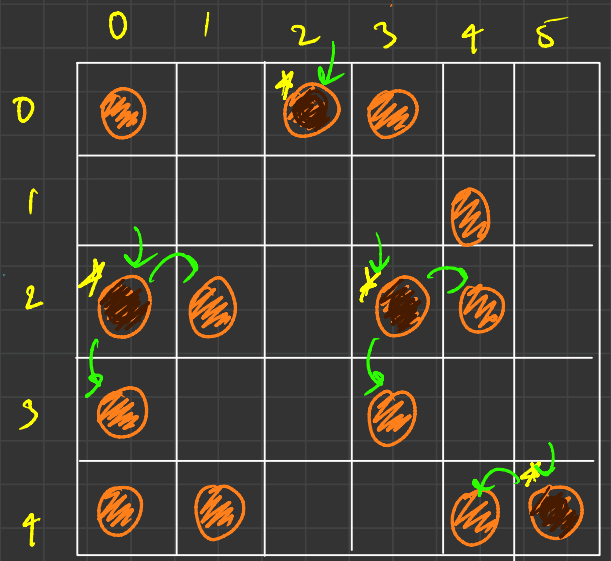
Rotten Oranges.

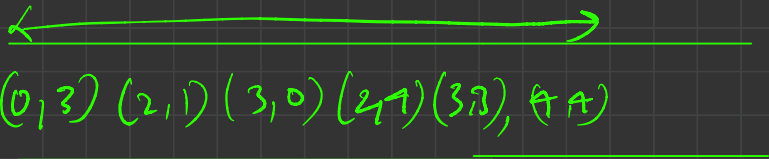
t=1

t=2

t=3

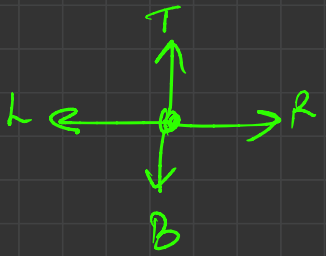
-1

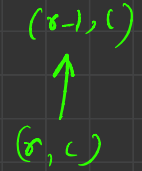




que

(4,0)





Boundry Traversal