Queue

Stack

↳ follows LIFO

→ Methods

1. push()
2. pop()
3. peek()
4. size()

} O(1)

---

queues

1. 10
2. 20
3. 30
4. 25

↓ seq of adding data

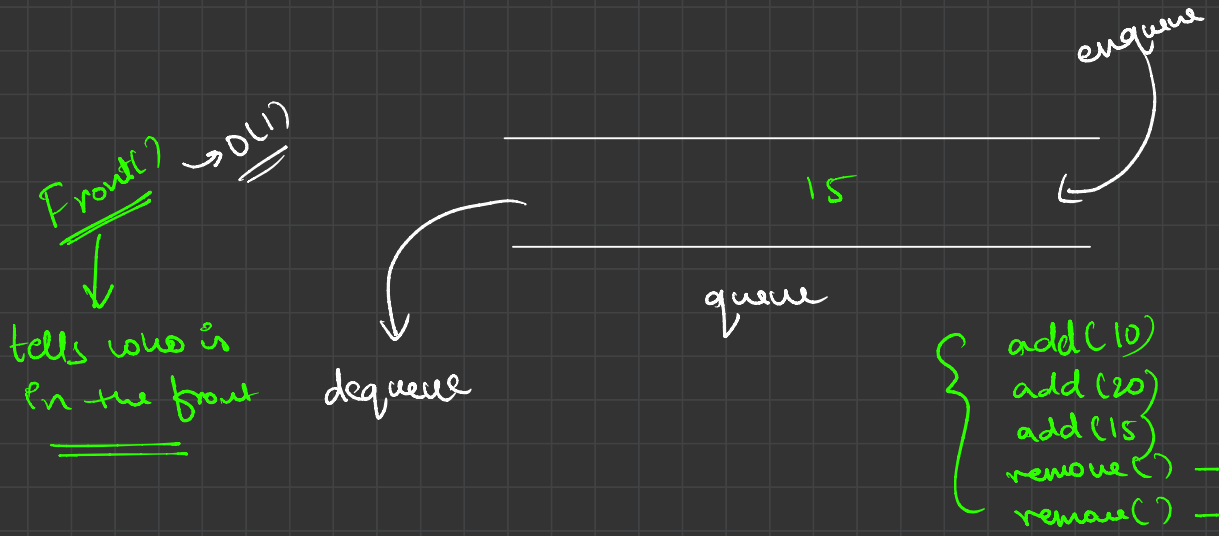
1. 10
2. 20
3. 30
4. 25

↓ seq. of Removal of data

FIFO → is followed by queue!

enqueue → Adding element in a queue. O(1)

dequeue → Removing element from a queue O(1)



front() → O(1)

tells value is in the front

add(10)
add(20)
add(15)
remove() → 10
remove() → 20

queue

- Linear Data structure
- follows FIFO (first In, first Out)

Methods → enqueue, dequeue, front(), size()

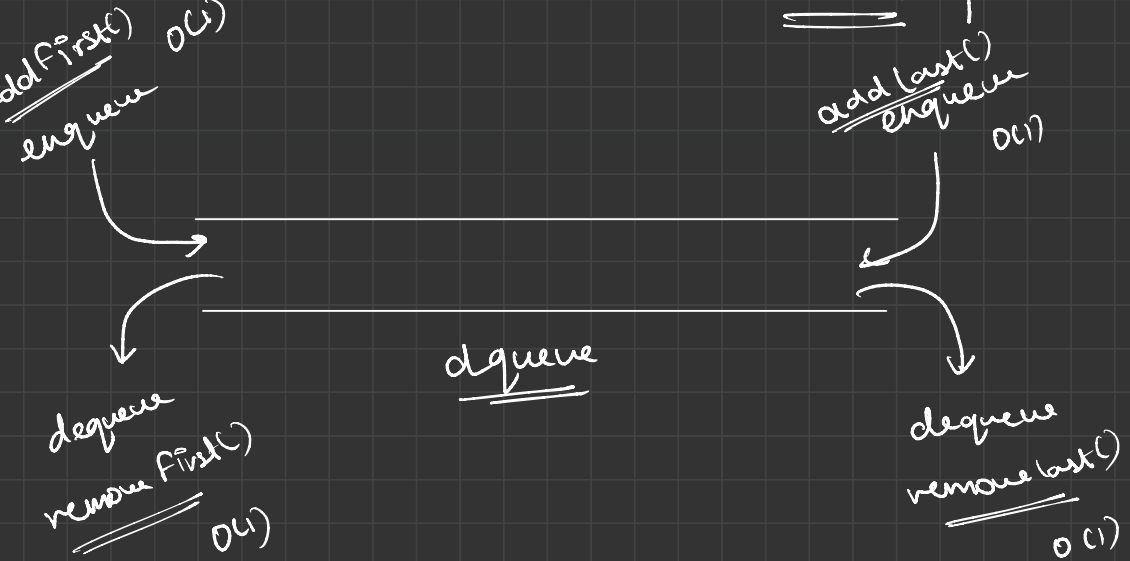
- enqueue → add()
- dequeue → remove()

Deque

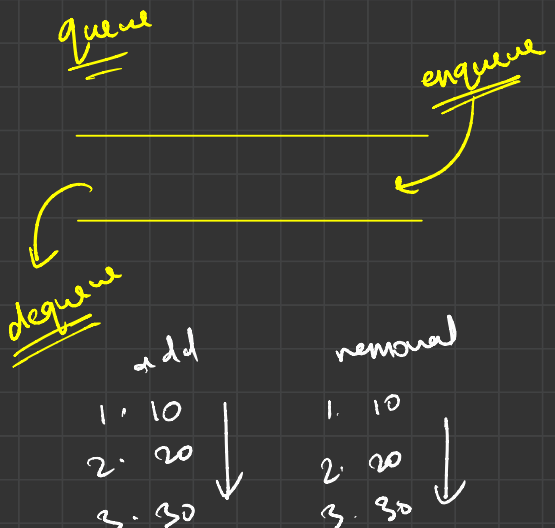
doubly ended Queue

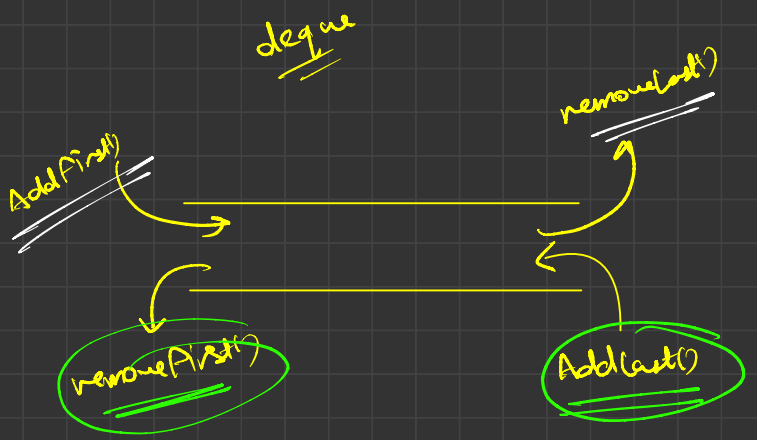
↳ uses doubly ended LinkedList internally.

linear ds!



Q Can you implement a queue using a deque ?

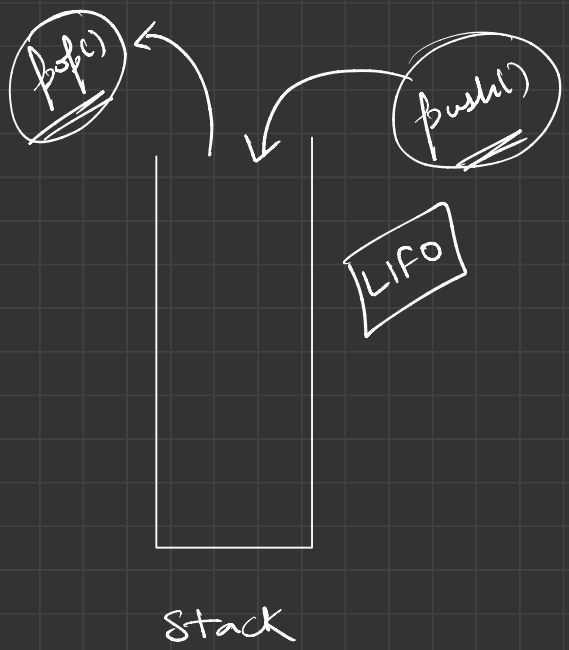


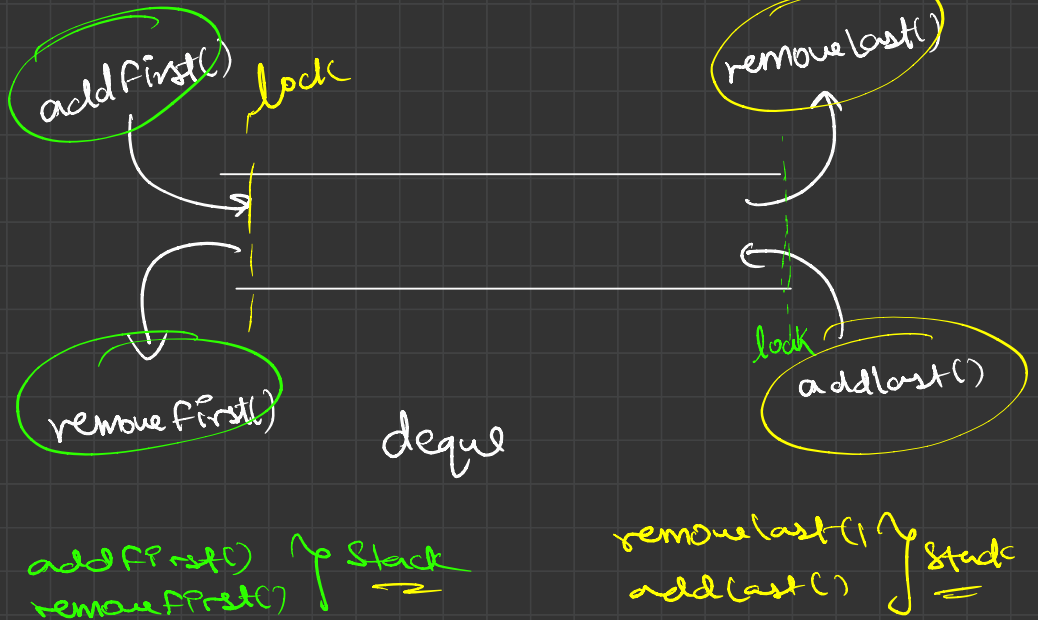


removeFirst()
AddLast()
} pair up to make FIFO operation.

AddFirst()
removeLast()
} pair up to make FIFO operations

Can you implement a stack using a deque?





addFirst()
removeFirst()
} Stack

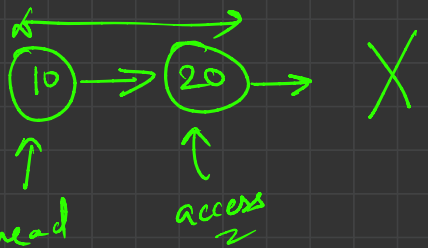
removeLast()
addLast()
} Stack

Design a stack using a linked list

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

Stack

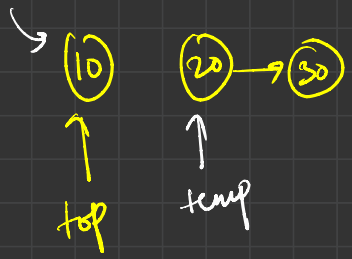
O(1) → add



O(N) → remove

O(1)
addFirst() → push()
removeFirst() → pop()
O(1)

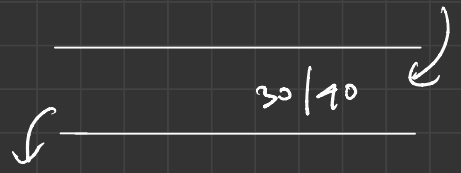
} Stack using LinkedList!

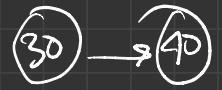


top.next = null

top = temp

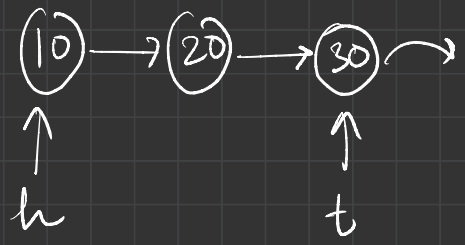
Implement a queue using a linked list



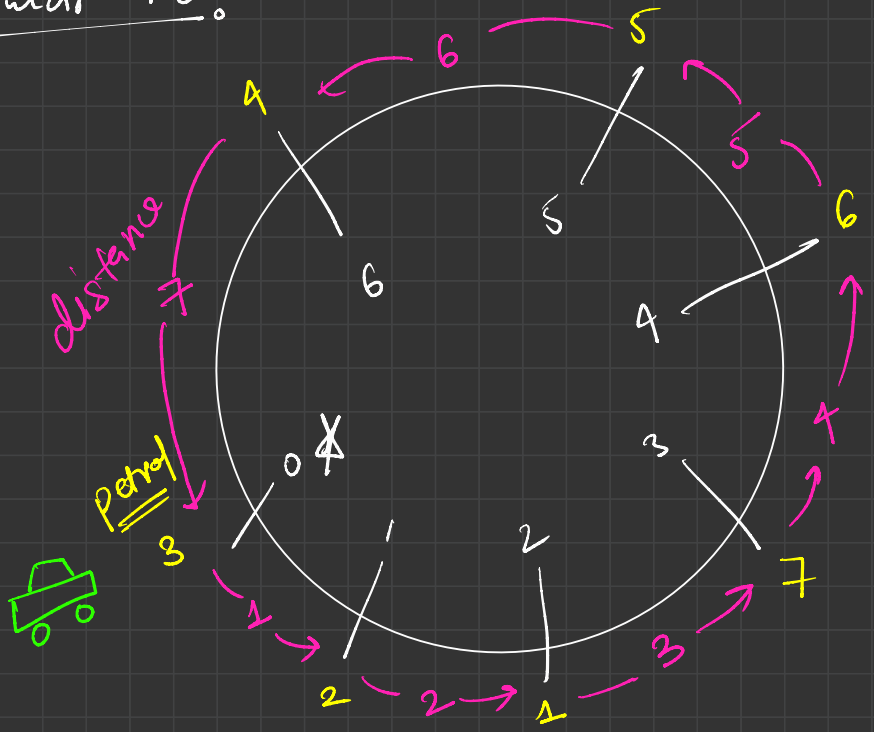


- add(10)
- add(20)
- add(30)
- add(40)
- remove() → 10
- remove() → 20

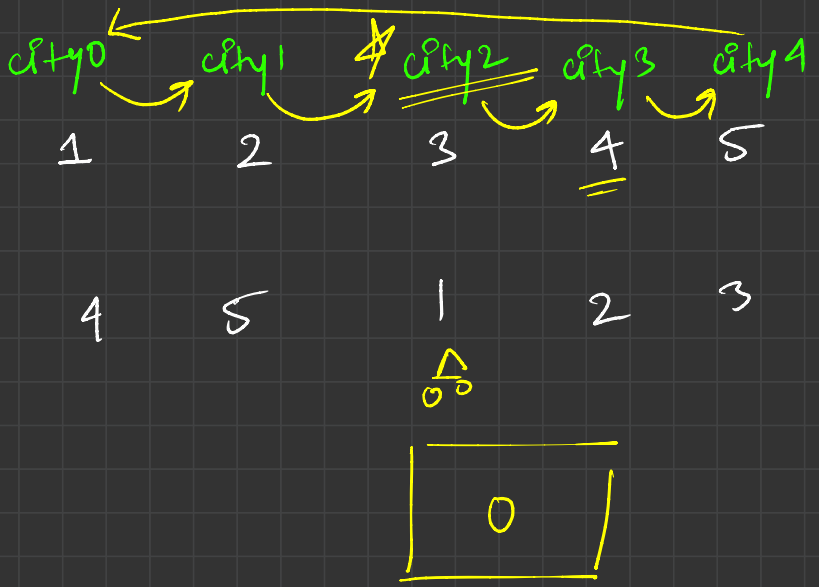
addLast()
removeFirst()
] queue

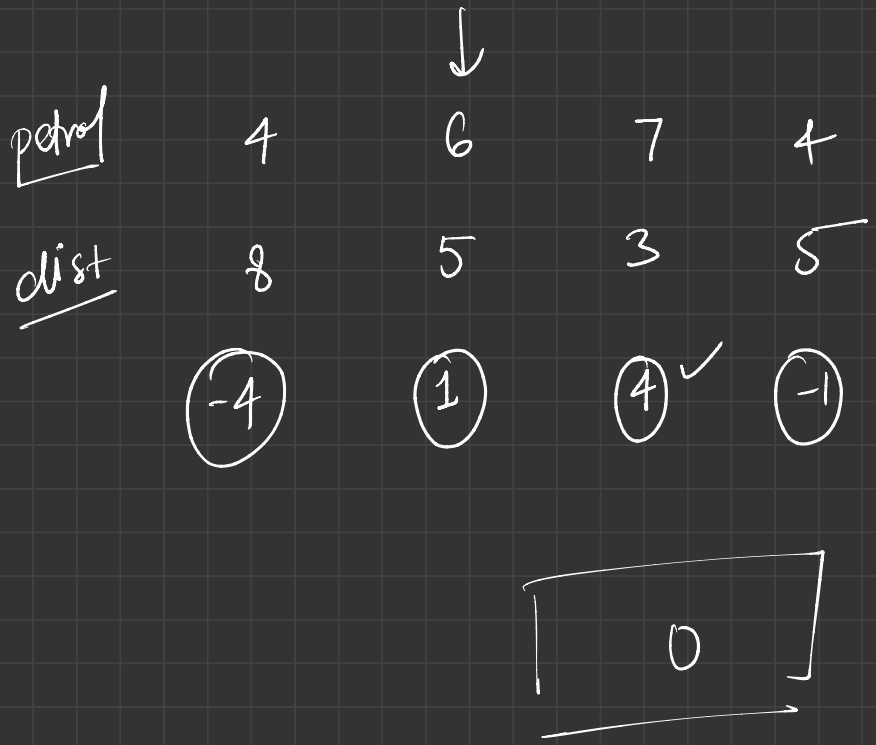

10
20
30
h
t

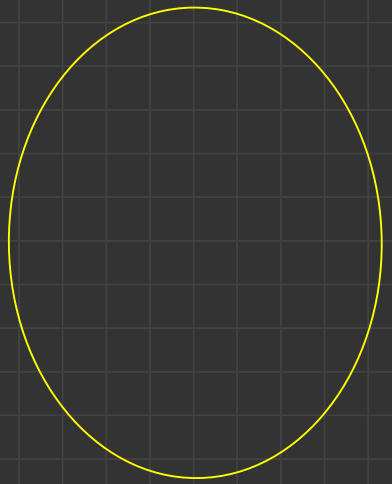
Circular Tour



fuel tank


city0
city1
city2
city3
city4
petrol
1
2
3
4
5
dist
4
5
1
2
3
0


petrol
4
6
7
4
dist
8
5
3
5
-4
1
4
-1
0



100km

↓

{100L of Petrol}

min

Step1

↳ check does we have minm fule req.

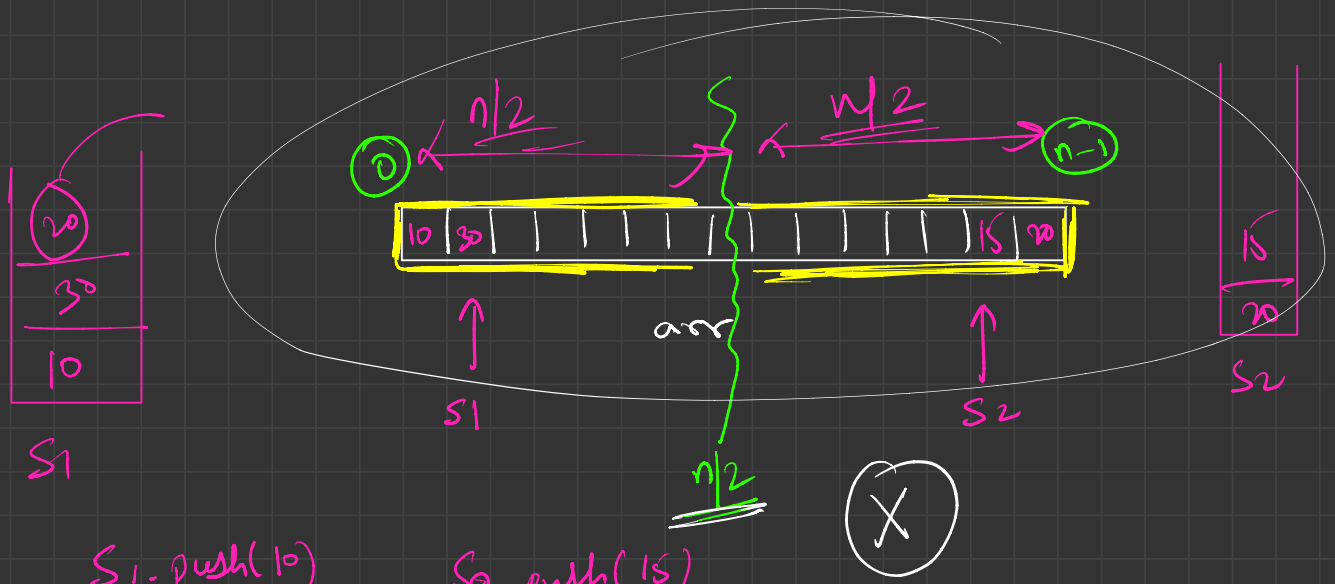
Atleast One solution will be present!

Petrol	4	6	7	4
dist	8	5	3	5

start = 1

remaining_petrol = 5 L

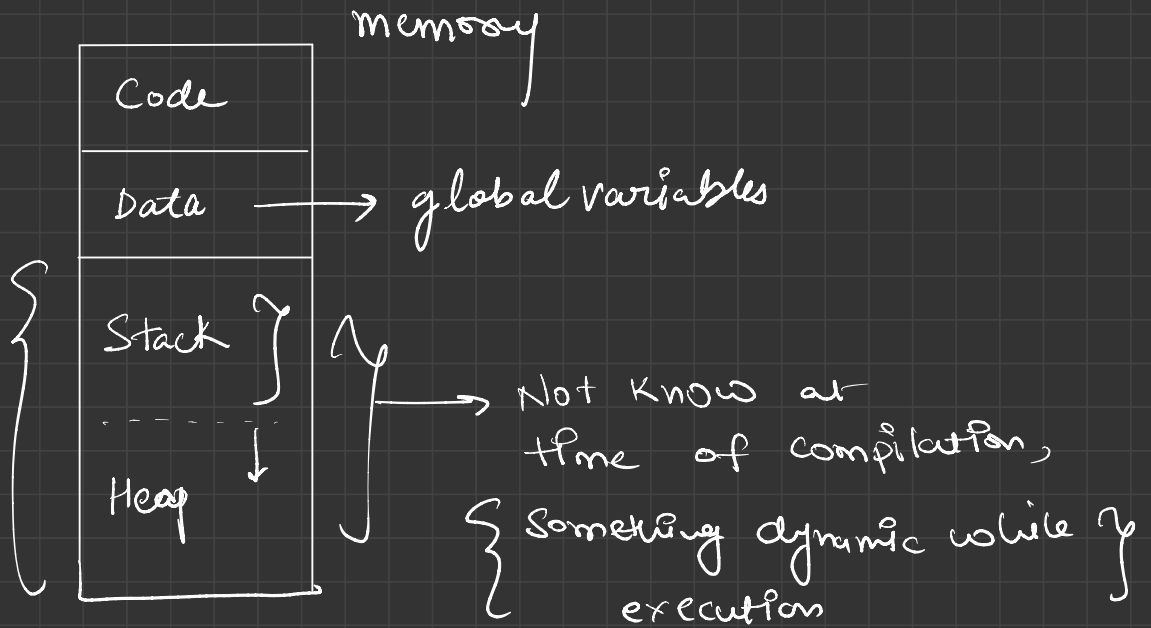
Implement two Stacks using an array!

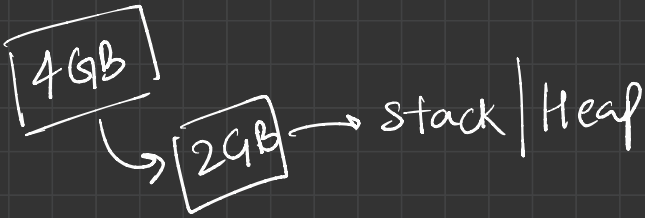


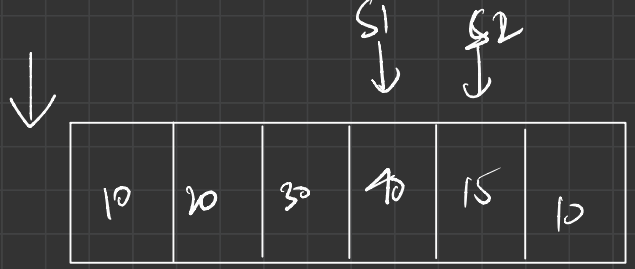
S1.push(10)
S2.push(20)
S1.push(30)
S1.push(20)

S2.push(15)
S1.pop() ⟶ 20

OS


memory
Code
Data
global variables
Stack
Heap
Not know at time of compilation,
Something dynamic while execution


4GB
2GB
stack | Heap



S1

push(10)

push(20)

push(30)

push(40)

pop()

S2

push(10)

push(15)

pop

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

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

// Method to pop an element from first stack
void pop1() {
  // Your code here
  if (top1 == -1) {
    System.out.println(-1);
  } else {
    int topValue = arr[top1];
    arr[top1] = 0;
    top1--;
    System.out.println(topValue);
  }
}

// Method to pop an element
// from second stack
void pop2() {
  // Your code here
  if (top2 == size)
    System.out.println(-1);
  } else {
    int topValue = arr[top2];
    arr[top2] = 0;
    top2++;
    System.out.println(topValue);
  }
}
```

TC
O(1)

TC
O(1)

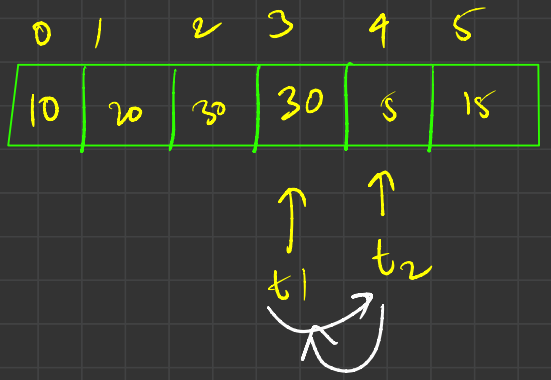
TC
O(1)

TC
O(1)

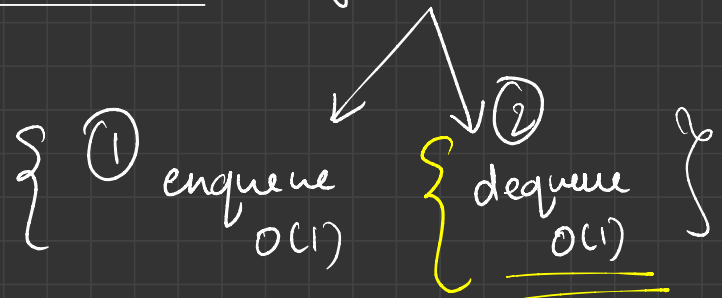
S1.push(30)

S2.push(40)

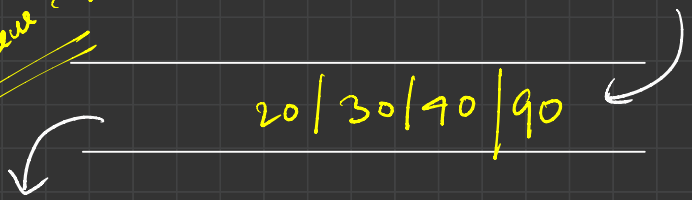
S1.push(35)



Implement Queue using Stacks (2 stacks)



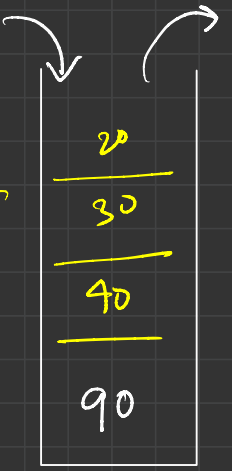
dequeue ~> O(1)



queue

dequeue ~> 10

Stack should have data in opp. fashion of insertion



S1

data

S2

supporting

add(90)

S1.pop()

10

Enqueue O(1)