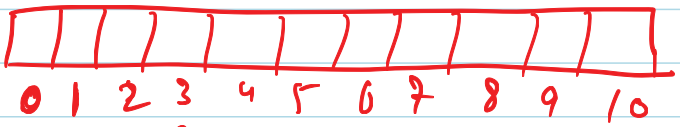
Stack & queue

① Array implementation of stack:

→ Capacity

→ arr

top



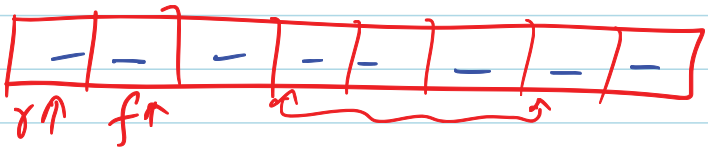
→ top-index = next available position.

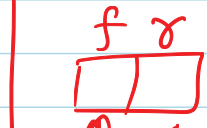
Empty → top == 0;
Stack full → top == capacity;

Push
arr[top++] = val

Pop
~~arr[top--] = 0;~~ ?
top--;

initial:
int f = 1
int r = 0
int* arr
int capacity





model

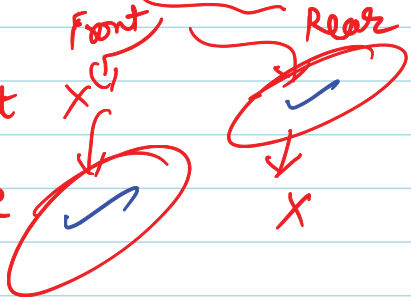
bool isFull() := f == 0 && capacity == rear;
bool isEmpty() :=

↳ HW → Array Implementation of queue

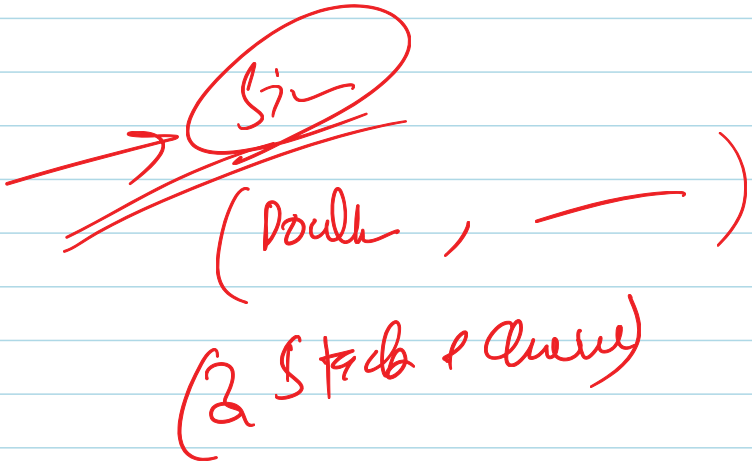
Design a stack using 2 queues ←

Design a queue using 2 stacks. ([push / pop] or on average, O(1) time)

one element can be moved atmost 3 times



n pop → O(1) Time O(n+m)
m push → O(1)



(2 Stacks & Queue)