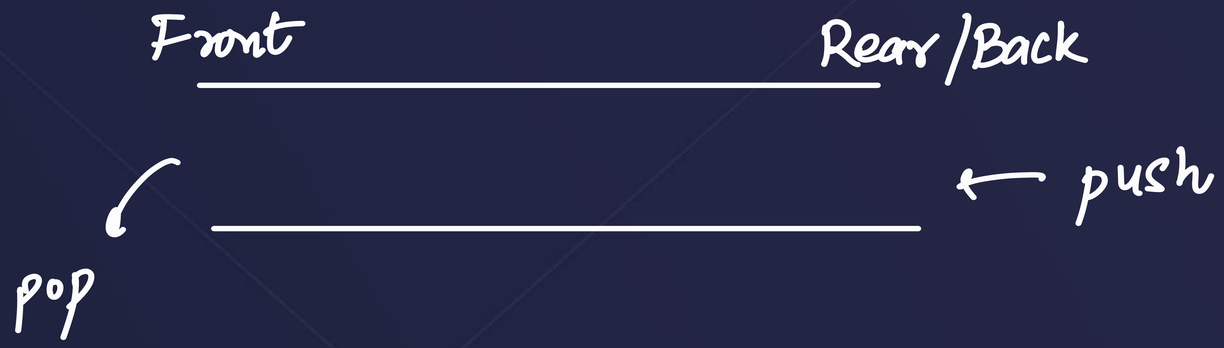
Queues -2

Lecture-50

Raghav Garg

Today's checklist

1) Interview Questions based on queues



Ques: Reverse first K elements of a Queue

K=2

f					b
3	4	5	6	2	1

f					b
2	1	3	4	5	6

Steps → for loop → K elements pop & push into stack

st me se pop & push to queue

Ques: Number of Students Unable to eat Lunch [LC - 1700]

Medium

Start end

{ 1 , 1 , 1 , 0 , 0 , 1 }

1 1 1

Stard en

{ 1 , 0 , 0 , 0 , 1 , 1 }

i

Count = 0 0 X X 0 0 X X 3

```
int count = 0;
while(q.size()>0 && count!=q.size()){
    if(q.front()==sandwiches[i]){
        count = 0; // VIMP
        q.pop();
        i++;
    }
    else{
        q.push(q.front());
        q.pop();
        count++; // VIMP
    }
}
return q.size();
```

Ques: Number of Students Unable to eat Lunch [LC - 1700]

{1, 1, 0, 0}

{0, 1, 0, 1}

i

Push Efficient approach

Ques: Implement Queue using Stacks [Leetcode - 232]

push → st.push( ) → O(1)

pop → remove & return the bottom ele of st

O(n)

O(n)

push

pop

front

back

2 stacks

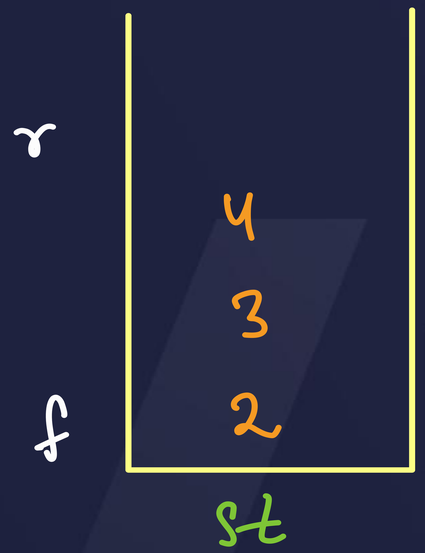
push(1)

push(2)

push(3)

push(4)

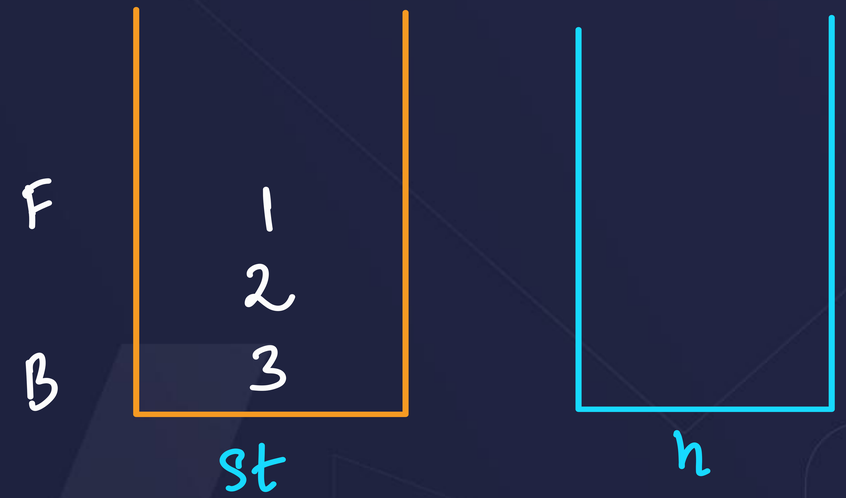
pop()



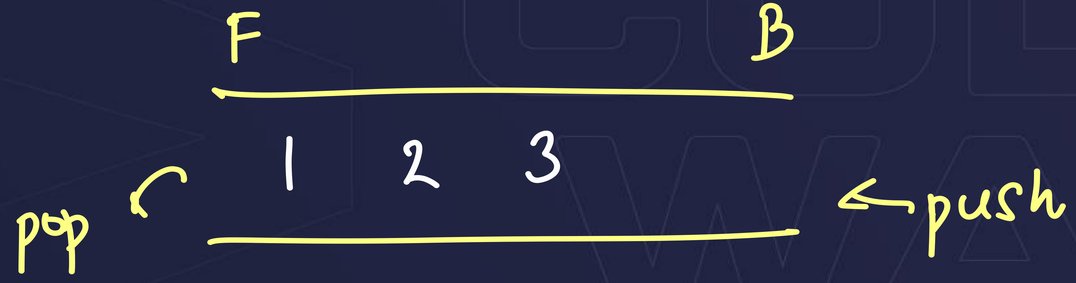
Ques: Implement Queue using Stacks [Leetcode - 232]

Pop - Efficient Approach

peek



push(1)
push(2)
push(3)
pop()
push(4)



Ques: First Negative in each window of size K

K = 3

arr = { 0, -1, -2, 3, 4, -5, 6, 4, 7, -8 }

n

ans = { -1, -1, -2, -5, -5, -5, 0, -8 }

Brute Force Sol^n

T.N.O : $(n-k+1)*k$

= T.C ⇒ $O(n*k)$

No. of windows = $n-k+1$

Ques: First Negative in each window of size K

0	1	2	3	4	5	6	7	8	9
arr = { 0 ,	-1 ,	-2 ,	3 ,	4 ,	-5 ,	6 ,	4 ,	7 ,	-8 }

i

i+K

ans = { -1, -1, -2, -5, -5, -5, 0, -8 }

THANK YOU