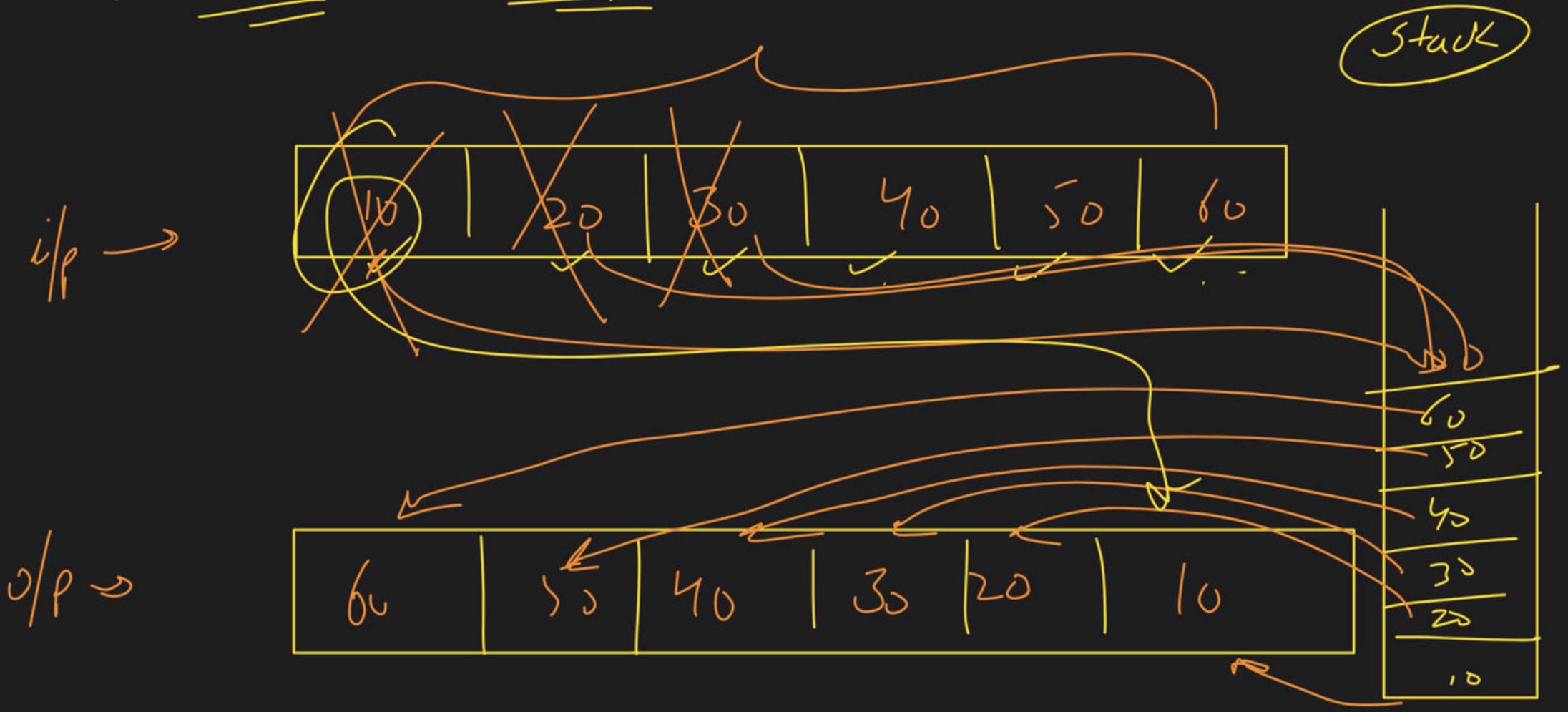
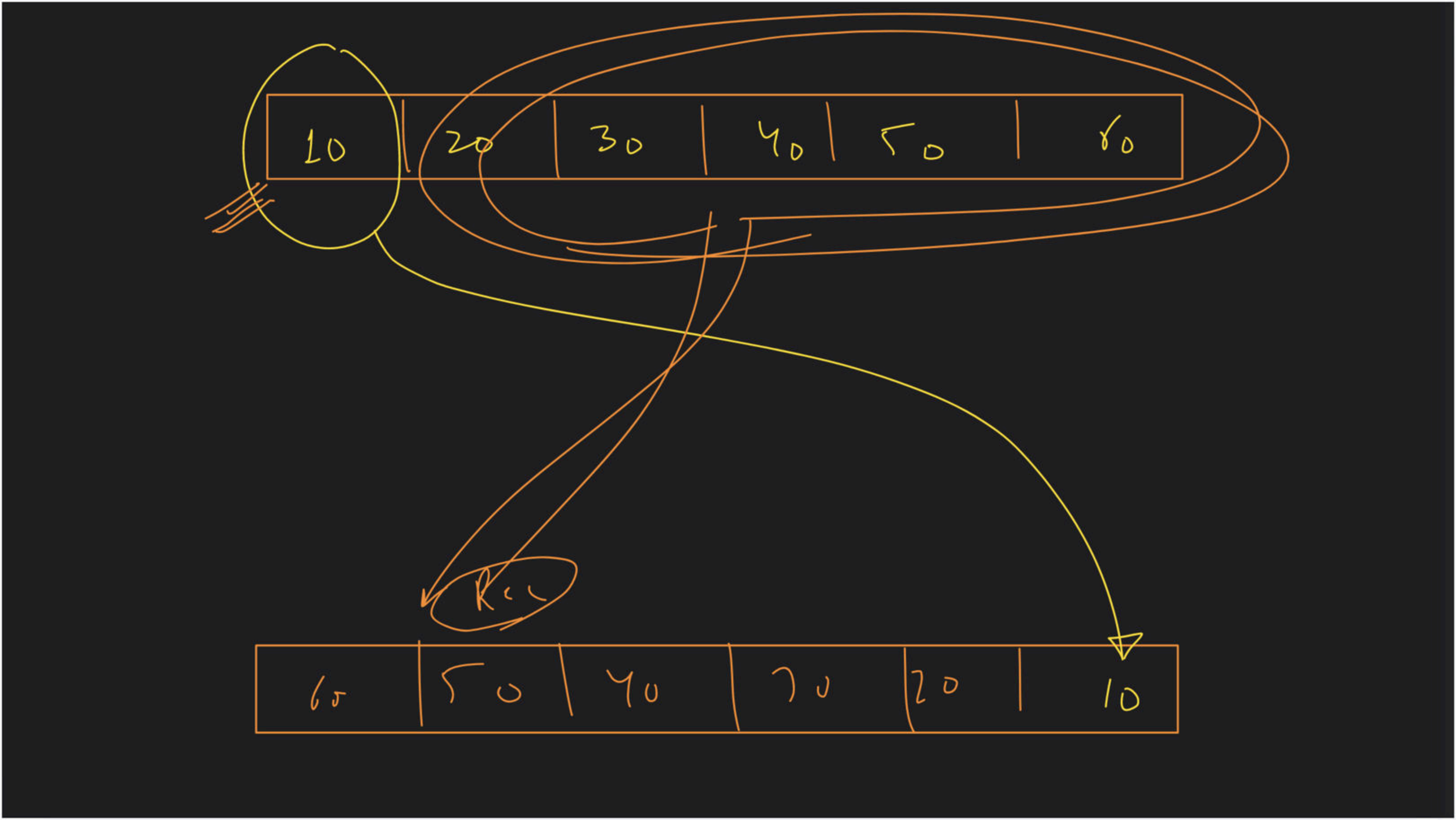
Queue Class - 2

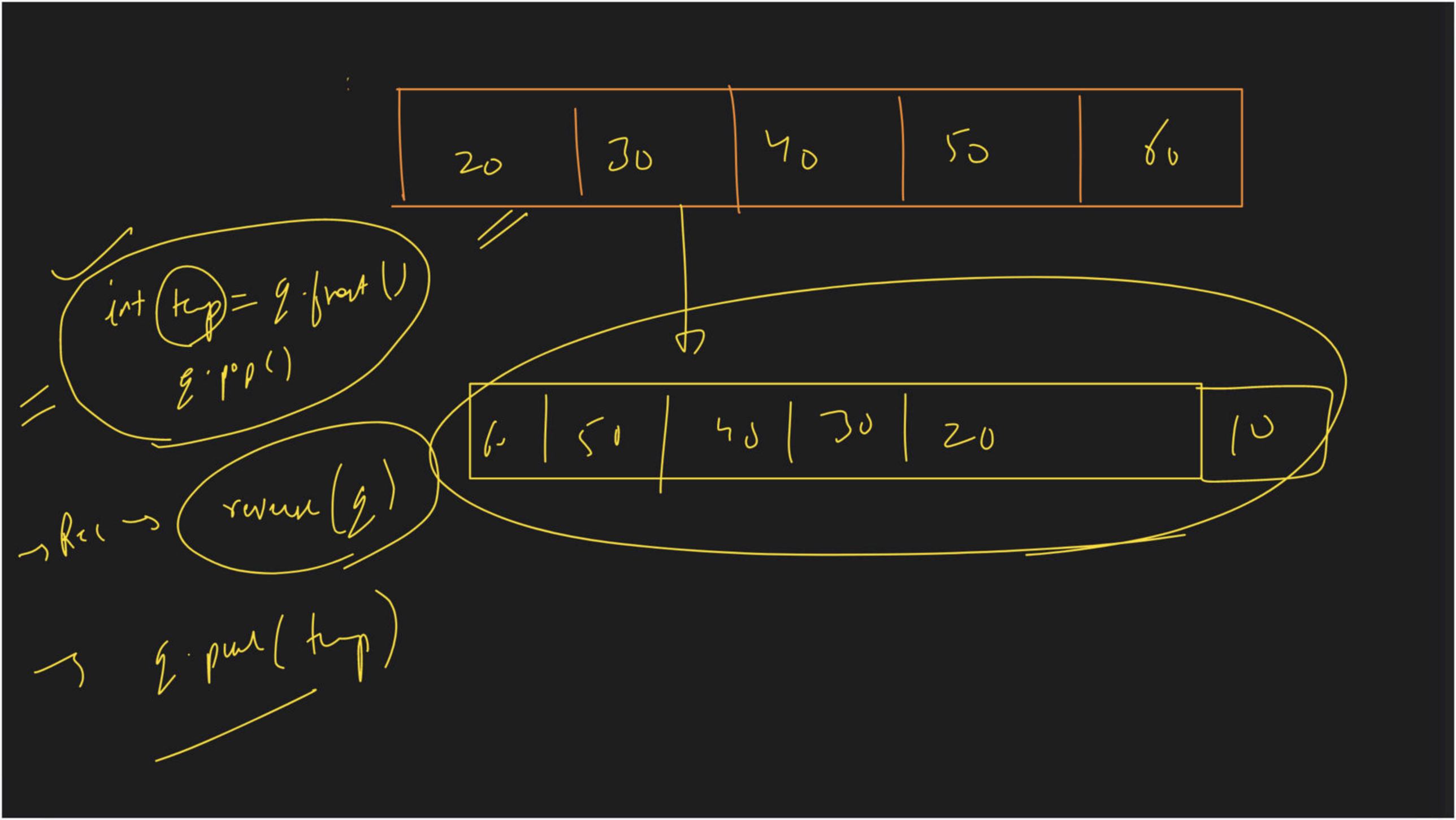
Special class

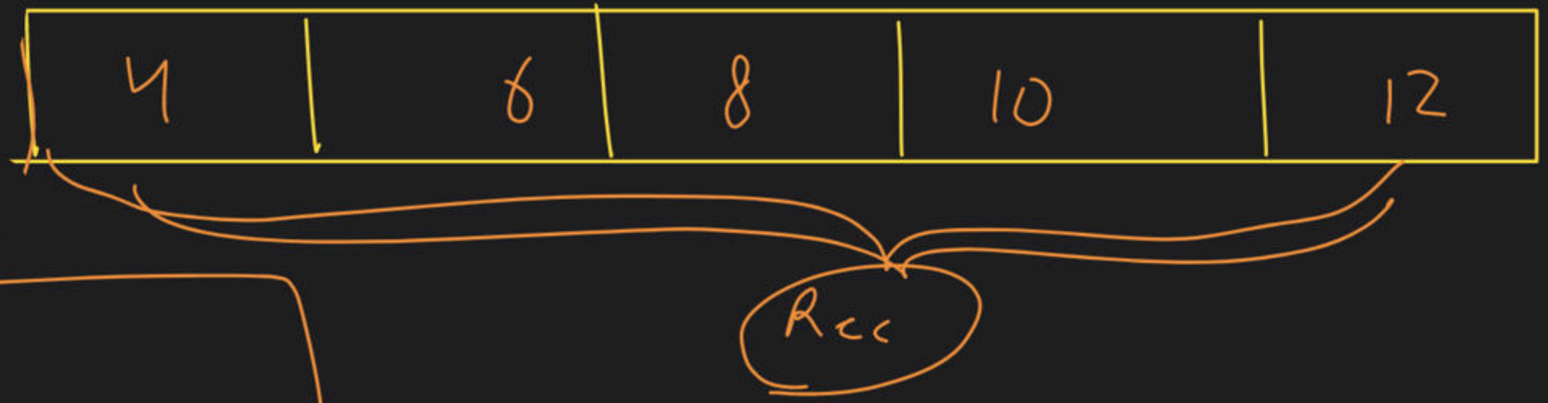
Love Babbar • Nov 22, 2023

→ Reverse a Queue



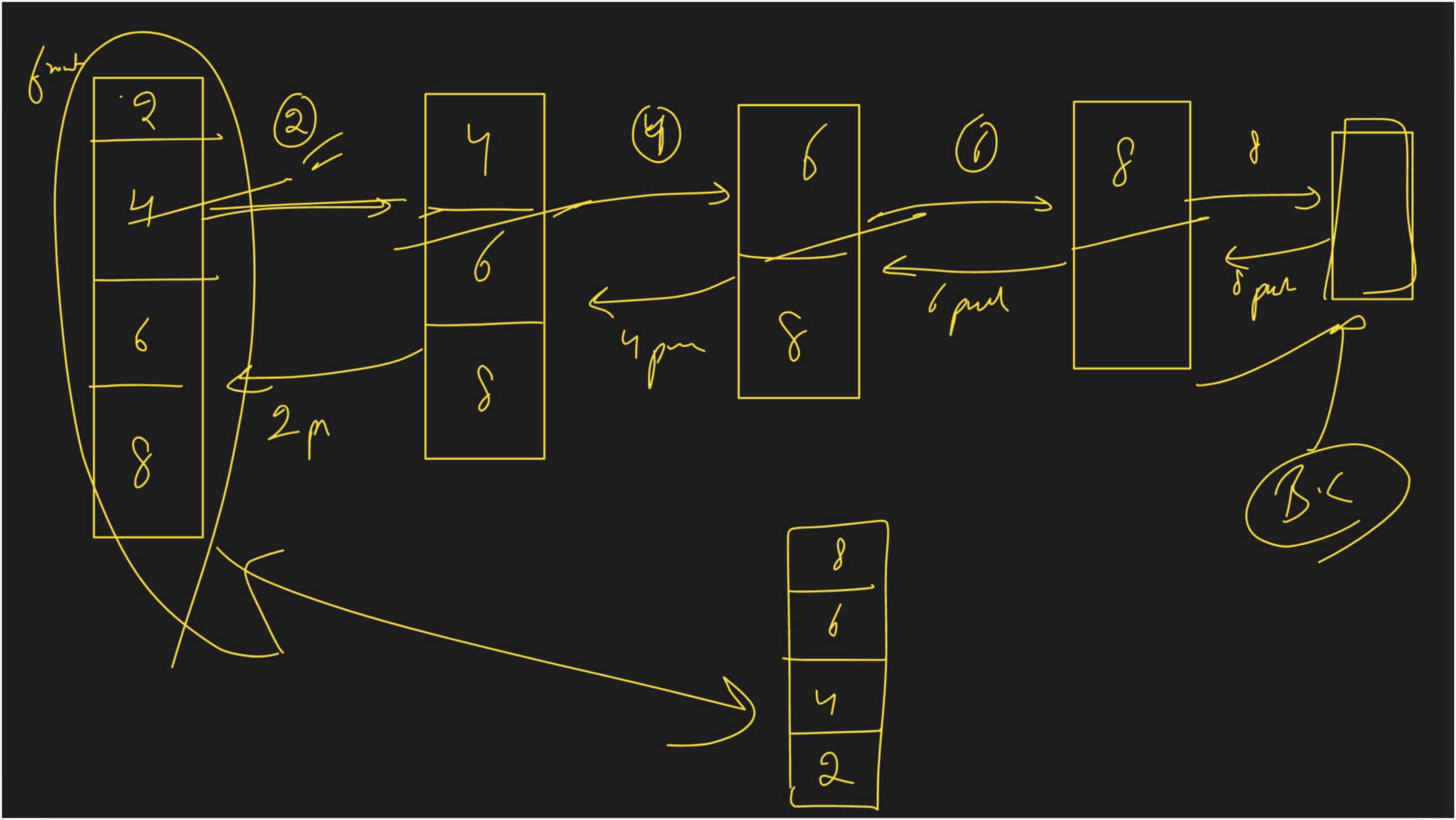

10
20
30
40
50
60
Rev
60
50
40
30
20
10

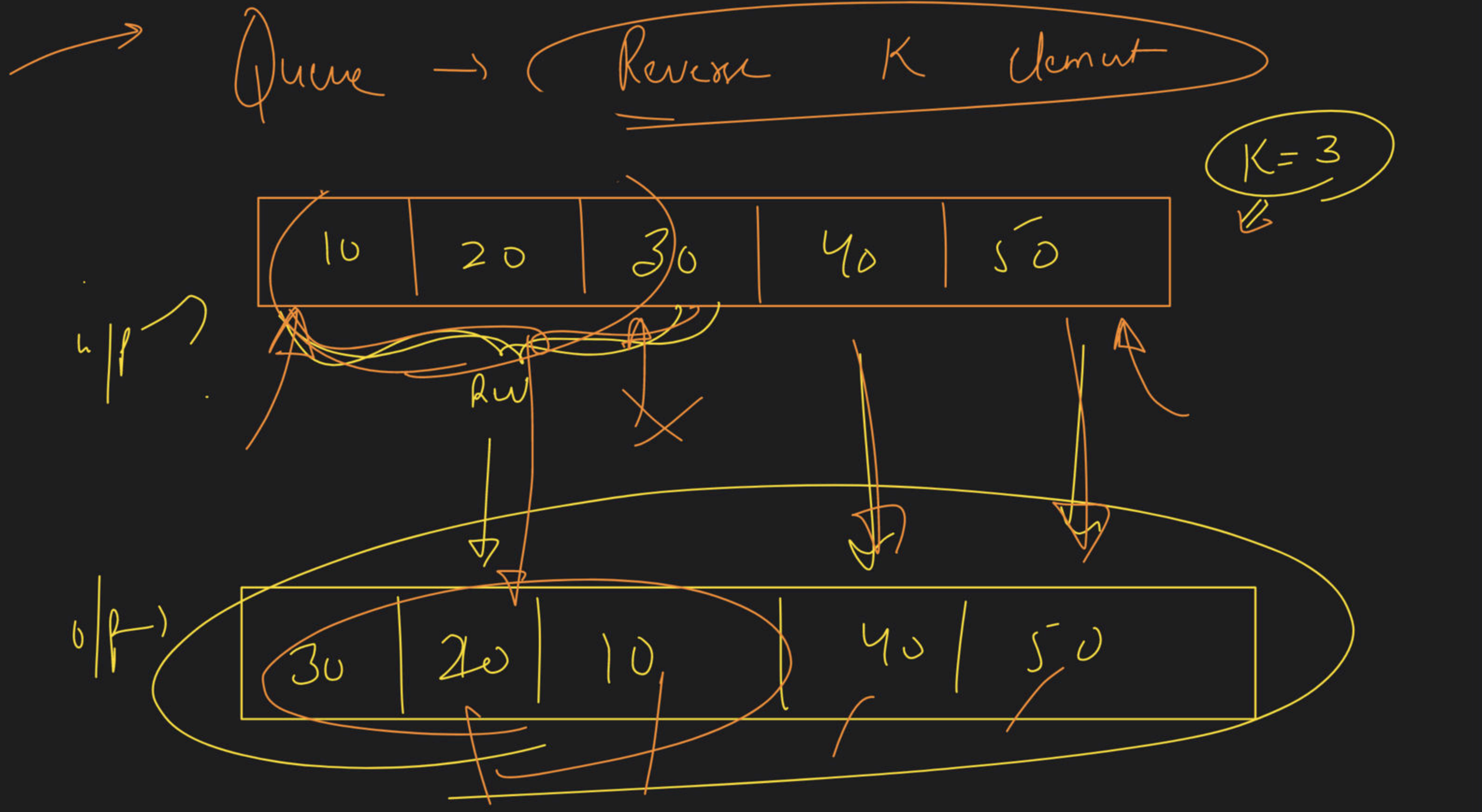

20
30
40
50
60
int temp = q.front()
q.pop()
=
return(q)
→ Rec →
→ q.push(temp)
60
50
40
30
20
10




```
int elemnt = q.front()
q.pop()
reverse(q)
q.push(elemnt)
```

elemnt = 2


2
4
6
8
2
4
6
8
4
6
8
6
8
8
2 pop
4 pop
6 pop
8 pop
B.C


Queue → Reverse K elements
K = 3
10
20
30
40
50
i/p
RW
X
o/p
30
20
10
40
50

#

(1) push first K elements from

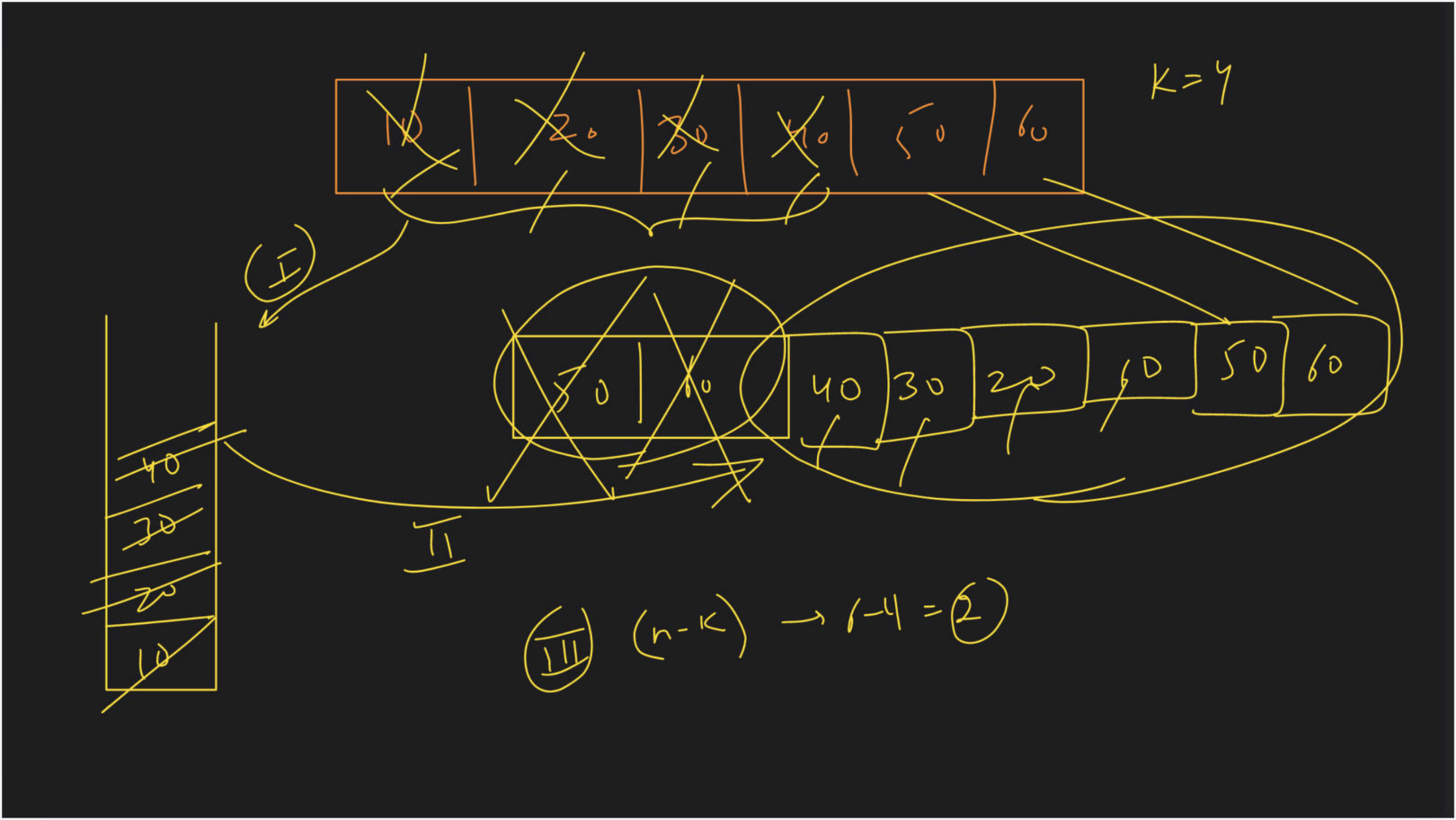
Q ⟶ S

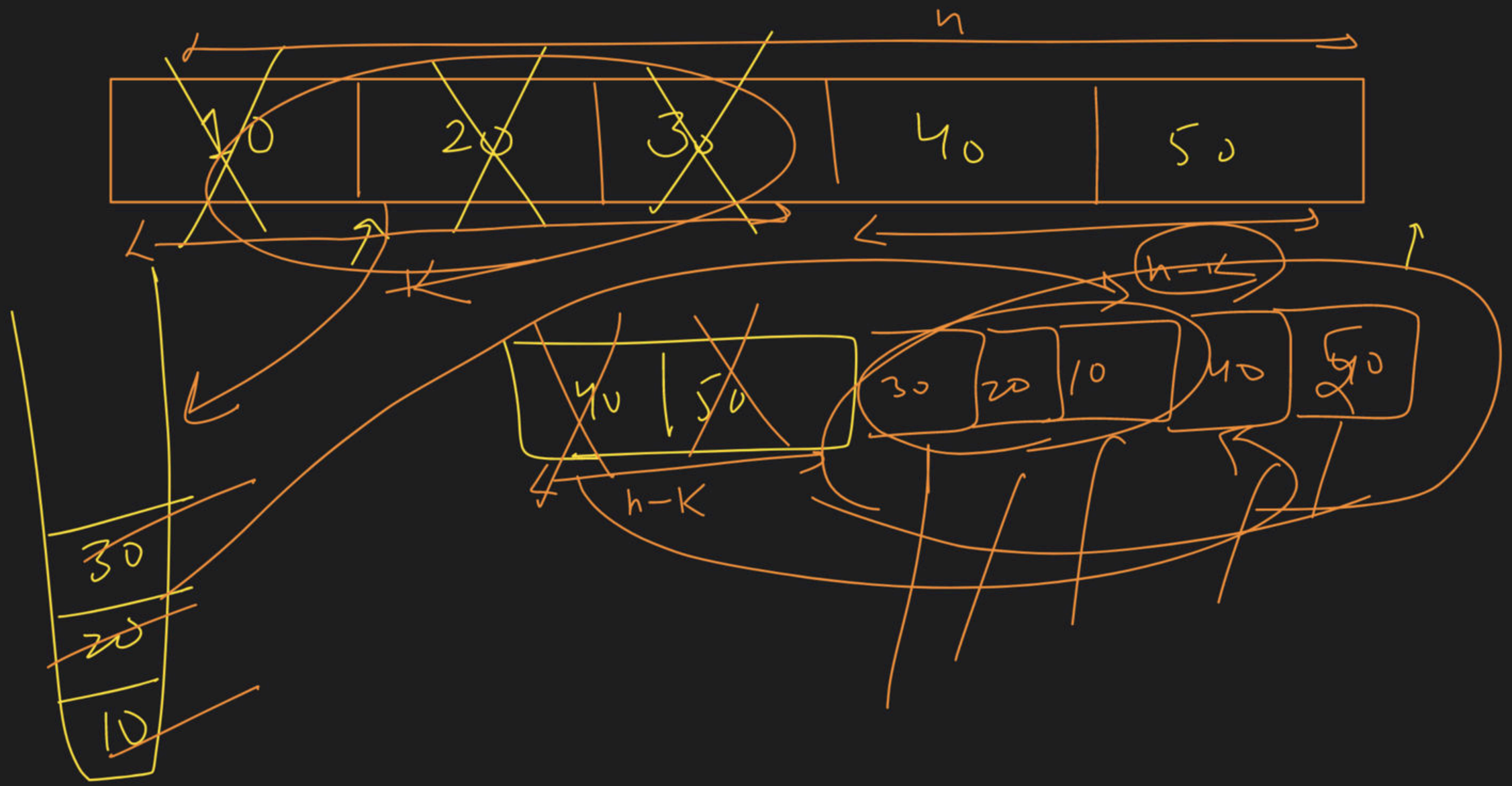
(2) push last K elements from

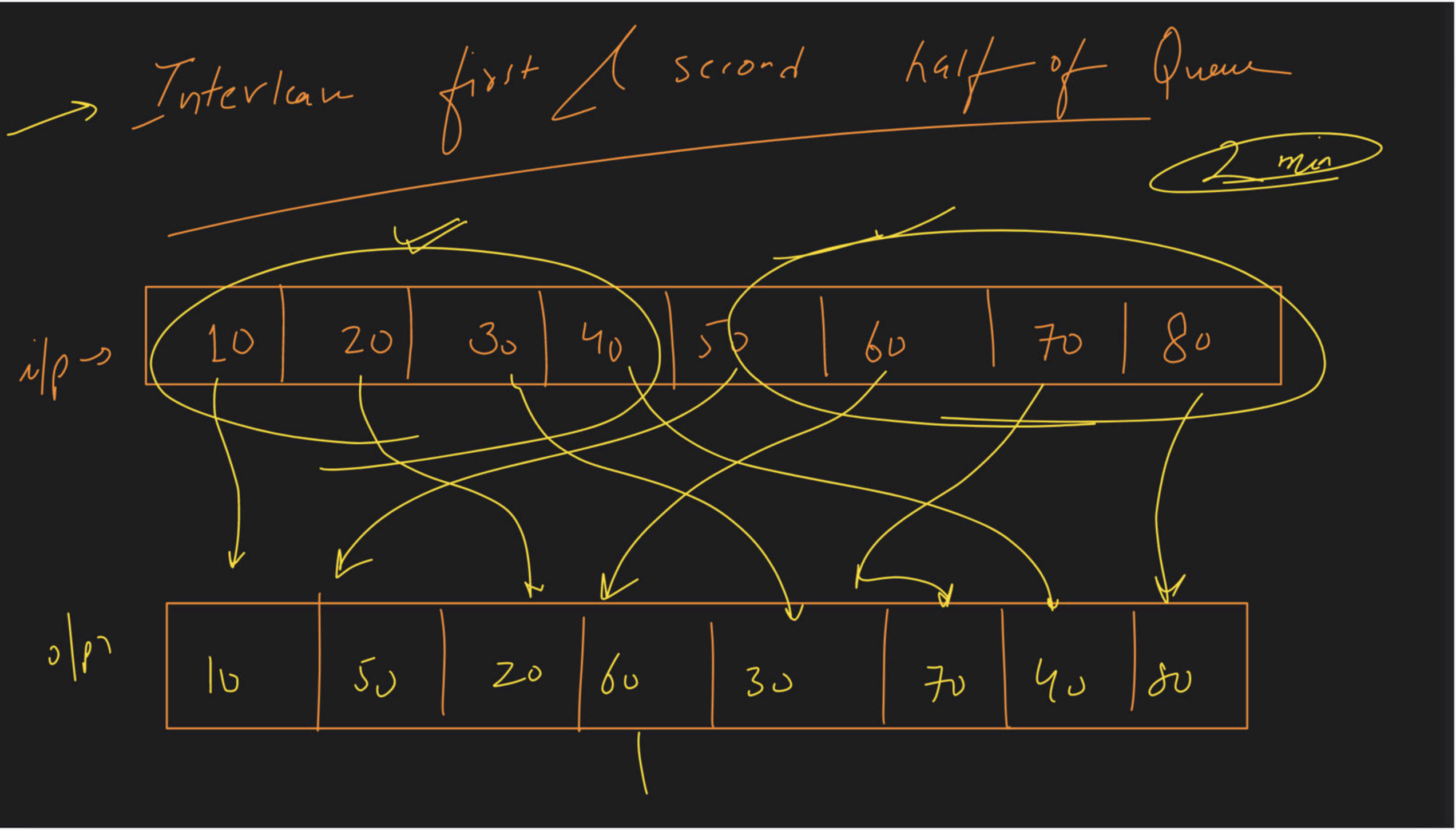
S ⟶ Q

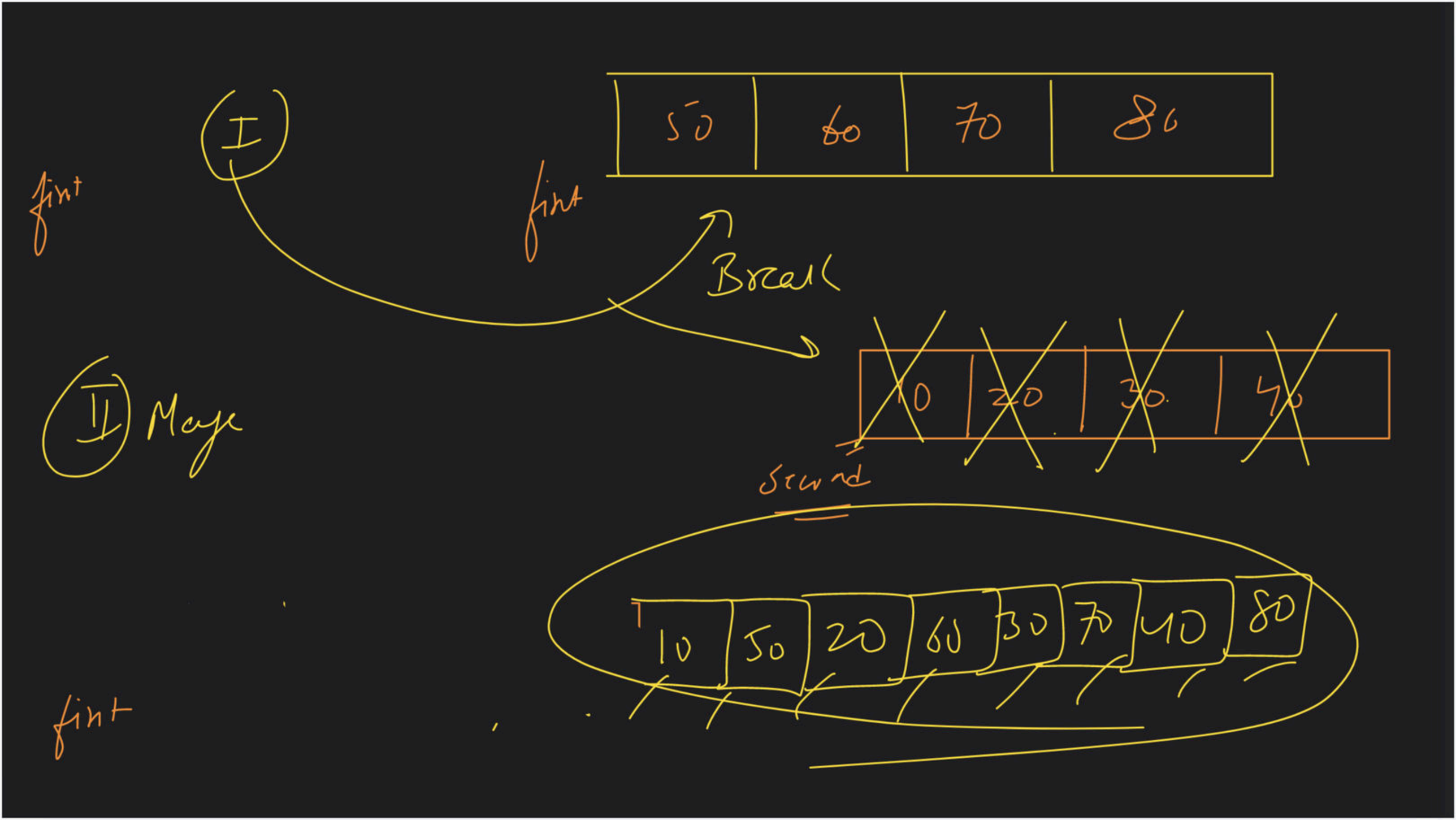
(3) Q ⟶ first (n − K)

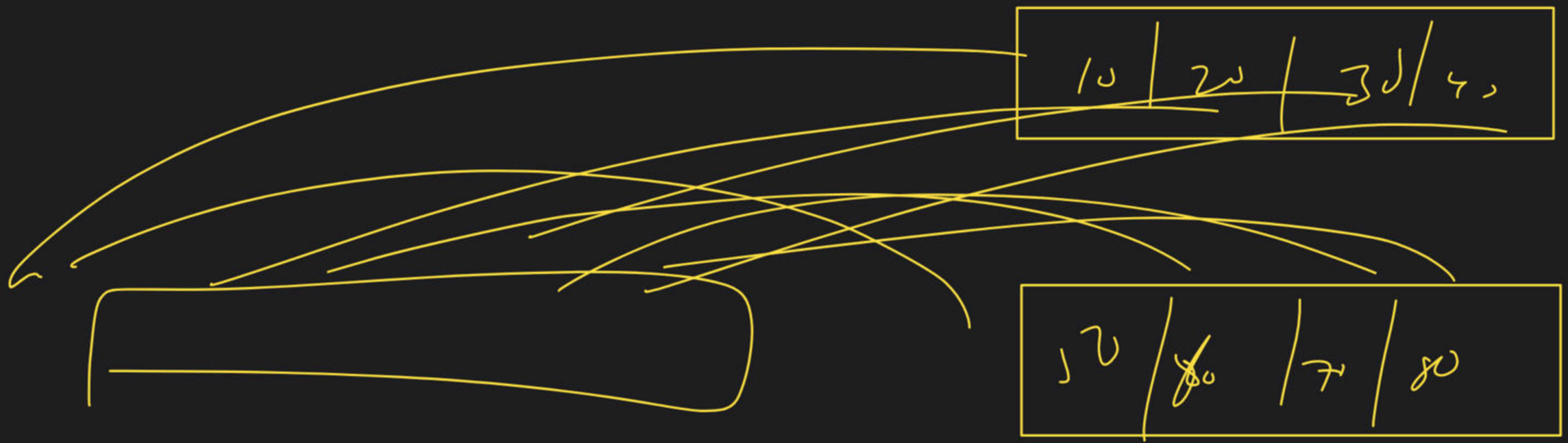
pop ⟶ push

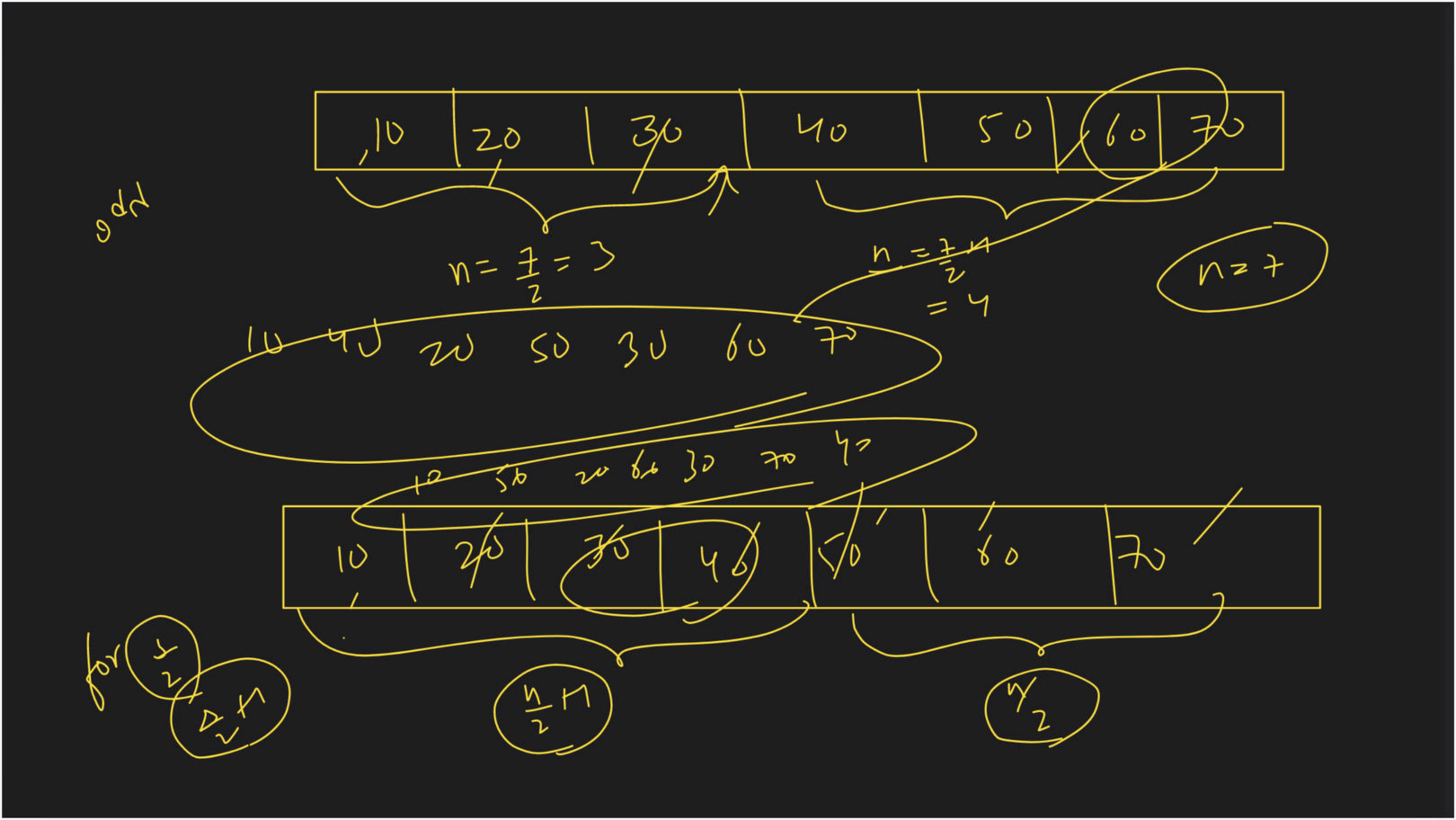

K=4
10
20
30
40
50
60
I
II
III
(n-k) → 6-4 = 2


n
10
20
30
40
50
k
n-k
40
50
30
20
10
40
50
n-k
30
20
10


Interleave first & second half of Queue
2 min
i/p
10
20
30
40
50
60
70
80
o/p
10
50
20
60
30
70
40
80


I
first
50
60
70
80
first
Break
II
Move
10
20
30
40
second
10
50
20
60
30
70
40
80
first



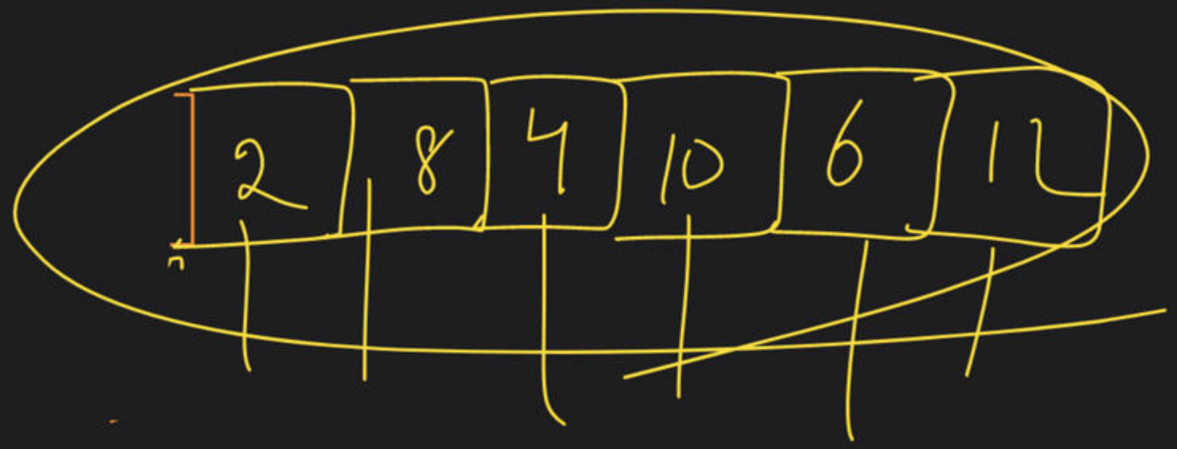

odd
10 20 30 40 50 60 70
n = 7/2 = 3
n = 7/2 = 4
n = 7
10 40 20 50 30 60 70
10 50 20 60 30 70 40
10 20 30 40 50 60 70
for
n/2
n/2 + 1
n/2 + 1
n/2

n = 6

k = 3

```
for (i=0; i<n/2; i++)
{
    int temp = first.front();
    first.pop();
    second.push(temp);
}
```

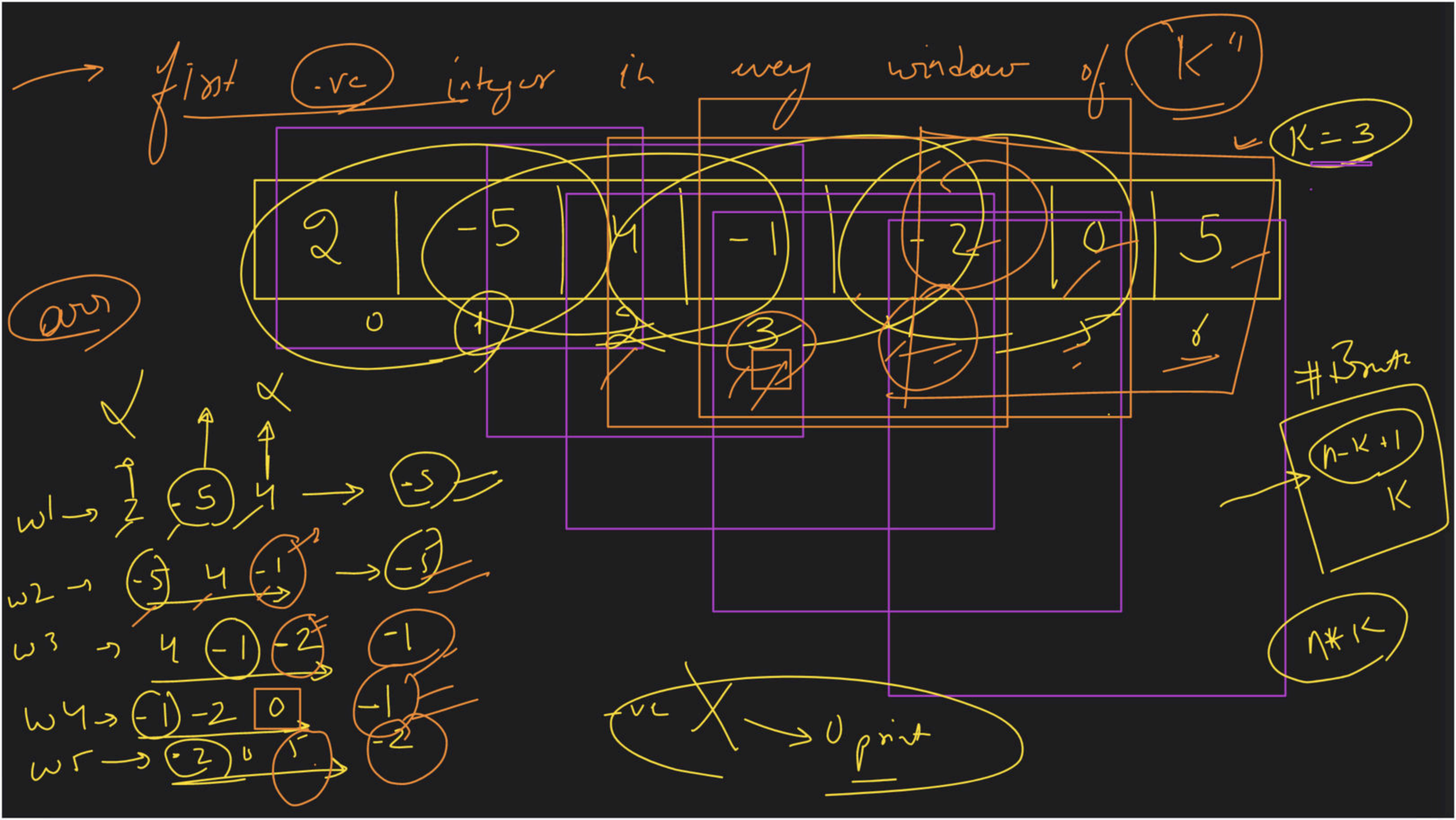
first

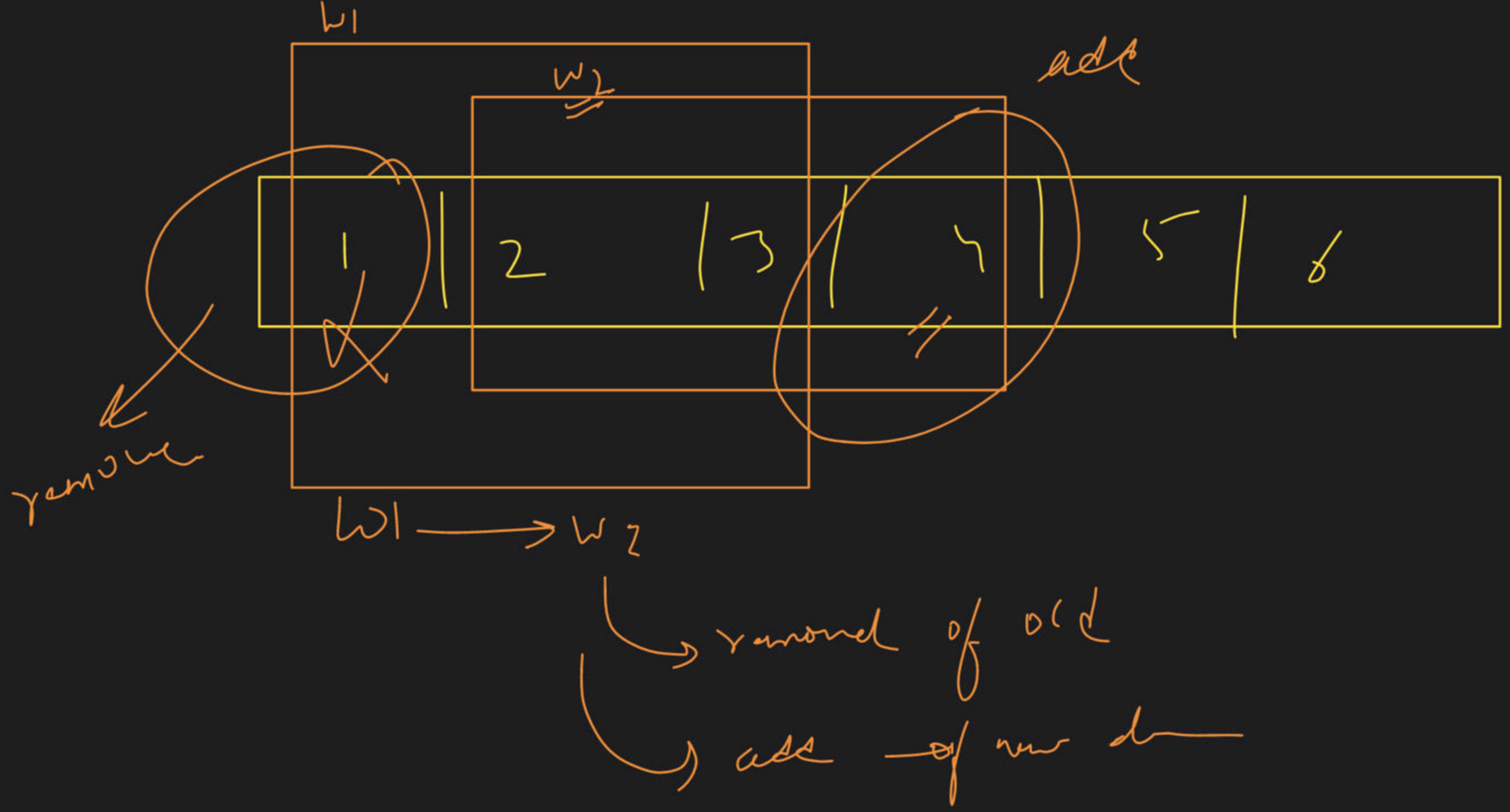


second

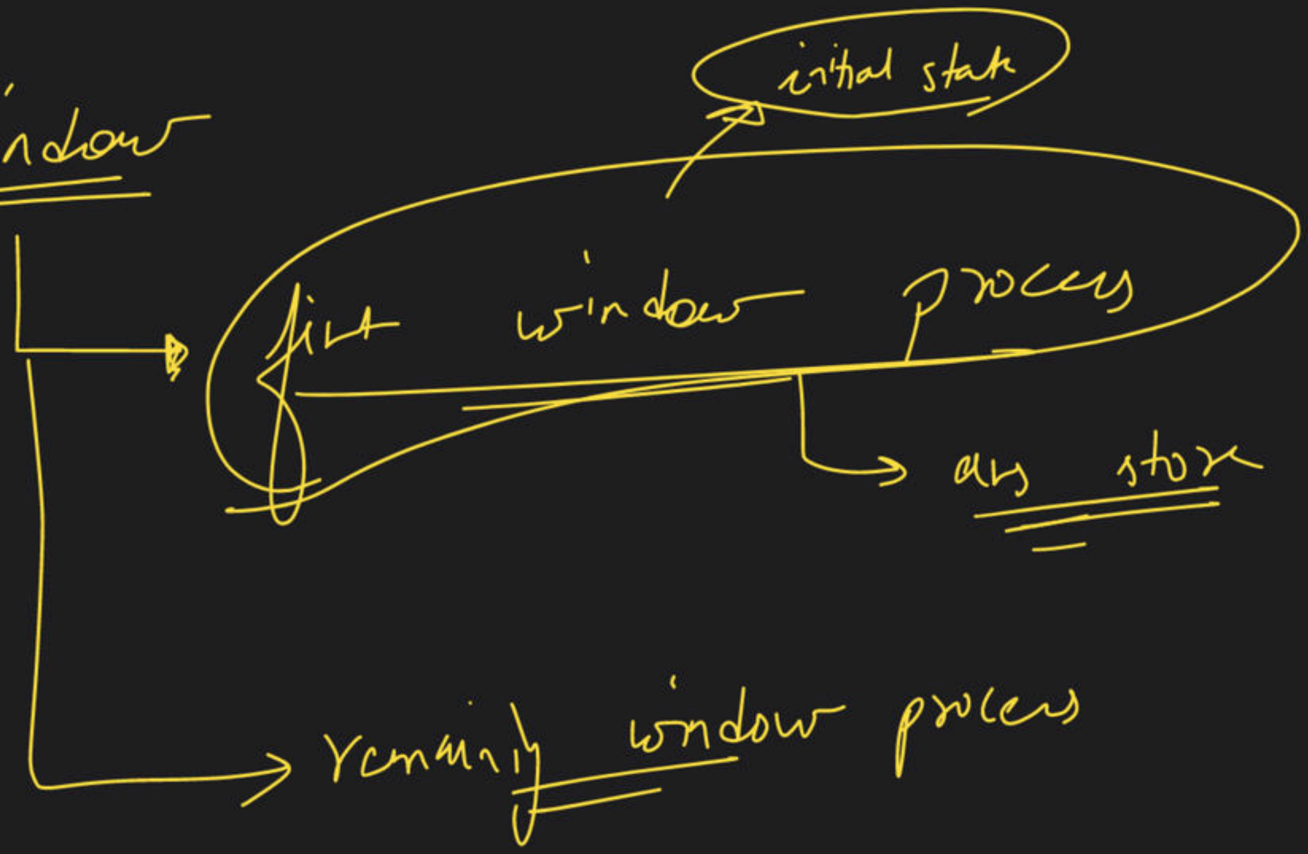

```
for (i=0; i<n/2; i++)
{
  tmp = second.front()
  second.pop()
  first.push(tmp)

  tmp = first.front()
  first.pop()
  first.push(tmp)
}
```

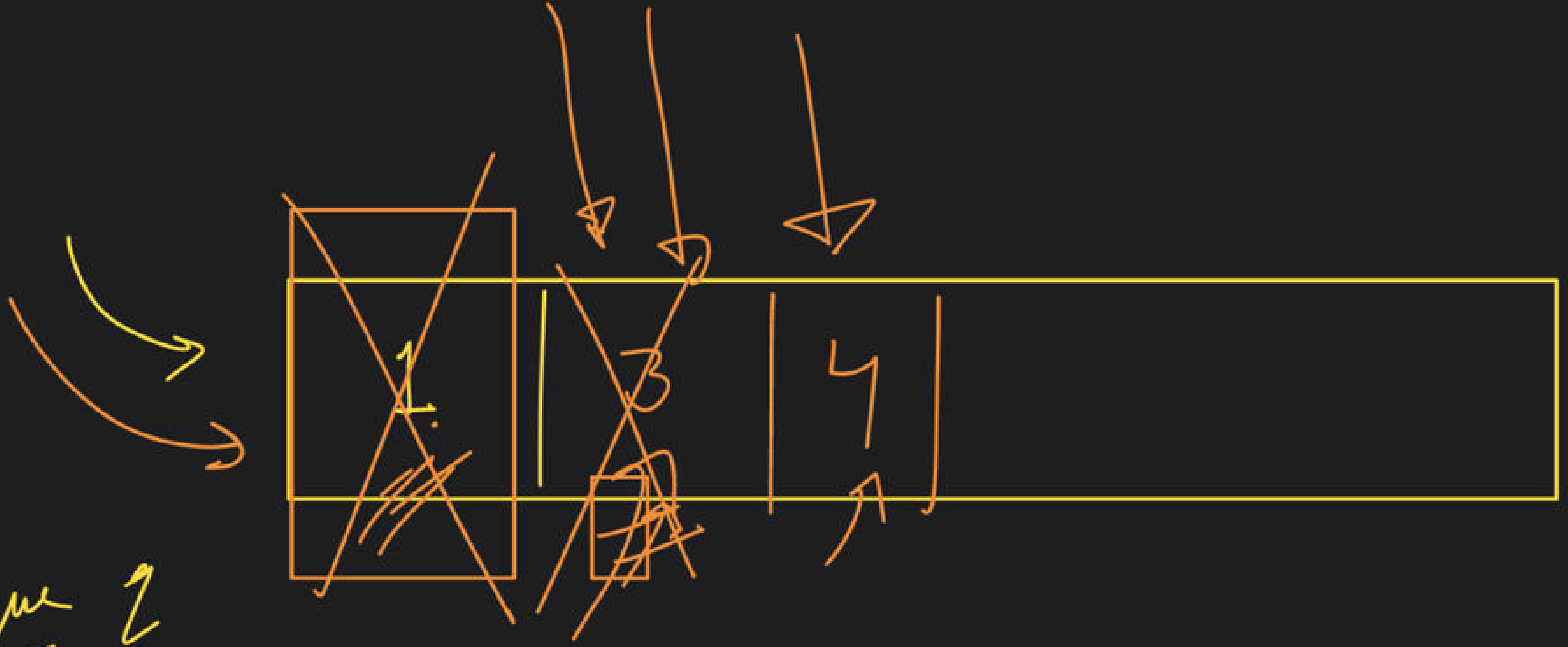

first -ve integer in every window of "k"
K=3
2 | -5 | 4 | -1 | -2 | 0 | 5
0 1 2 3 4 5
ans
w1 → -5
w2 → -5
w3 → -1
w4 → -1
w5 → -2
#Brute
n-k+1
k
n*k
-ve → 0 print


W1
W2
1
2
3
4
5
6
W1 → W2
removed of old
add of new

→ Sliding Window



degree



dequeue 2

index of -ve elmt

why

ans → 1 index → -5

→ -1

3 → -1

3 → -1

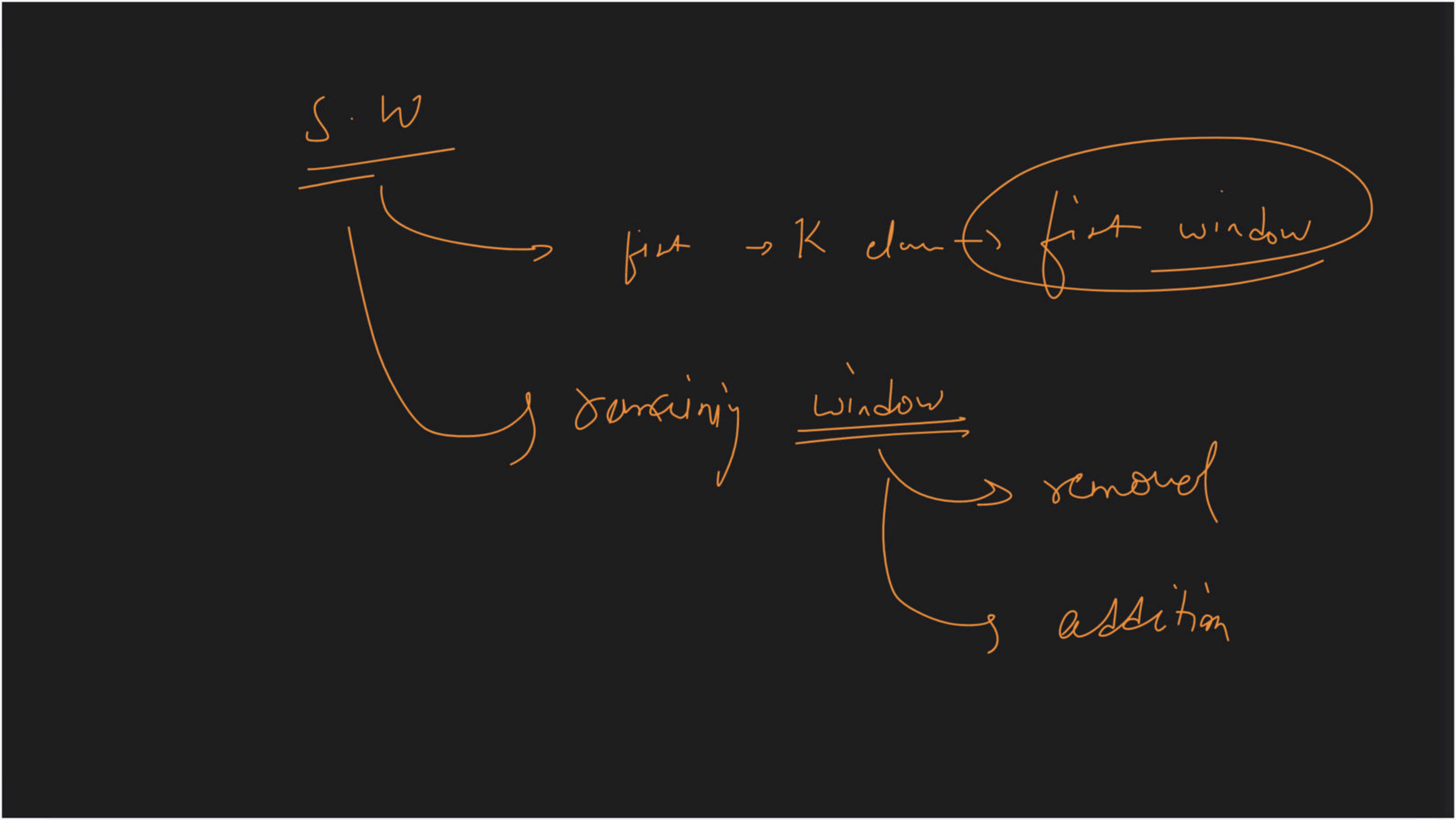
remove

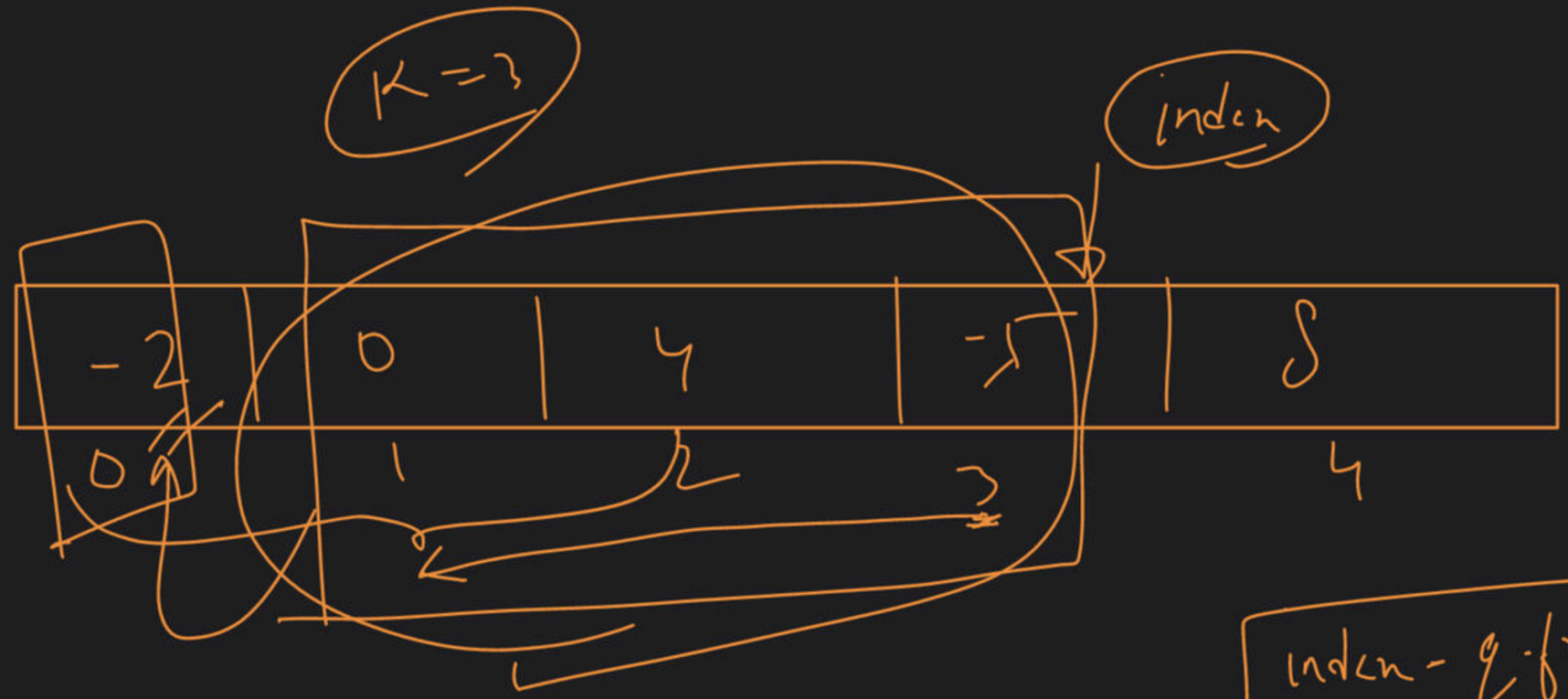
4 → -2

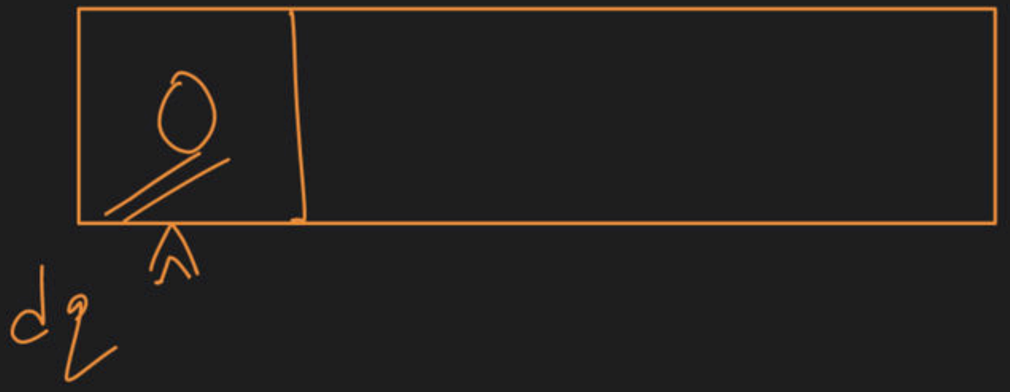
→ w2

w3

→ w4


S.W
fixed → K slide → fixed window
sensing window
removed
addition



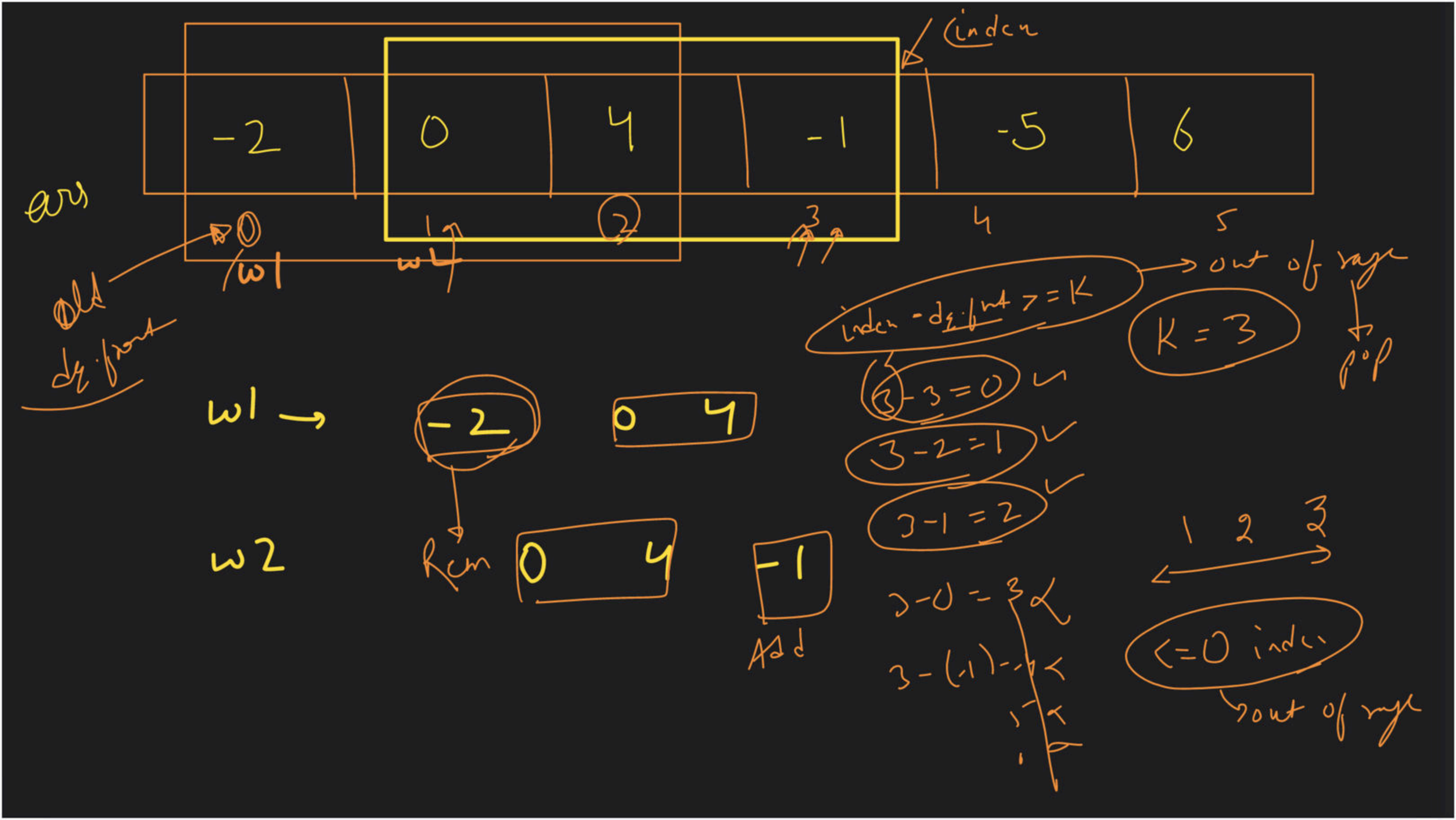


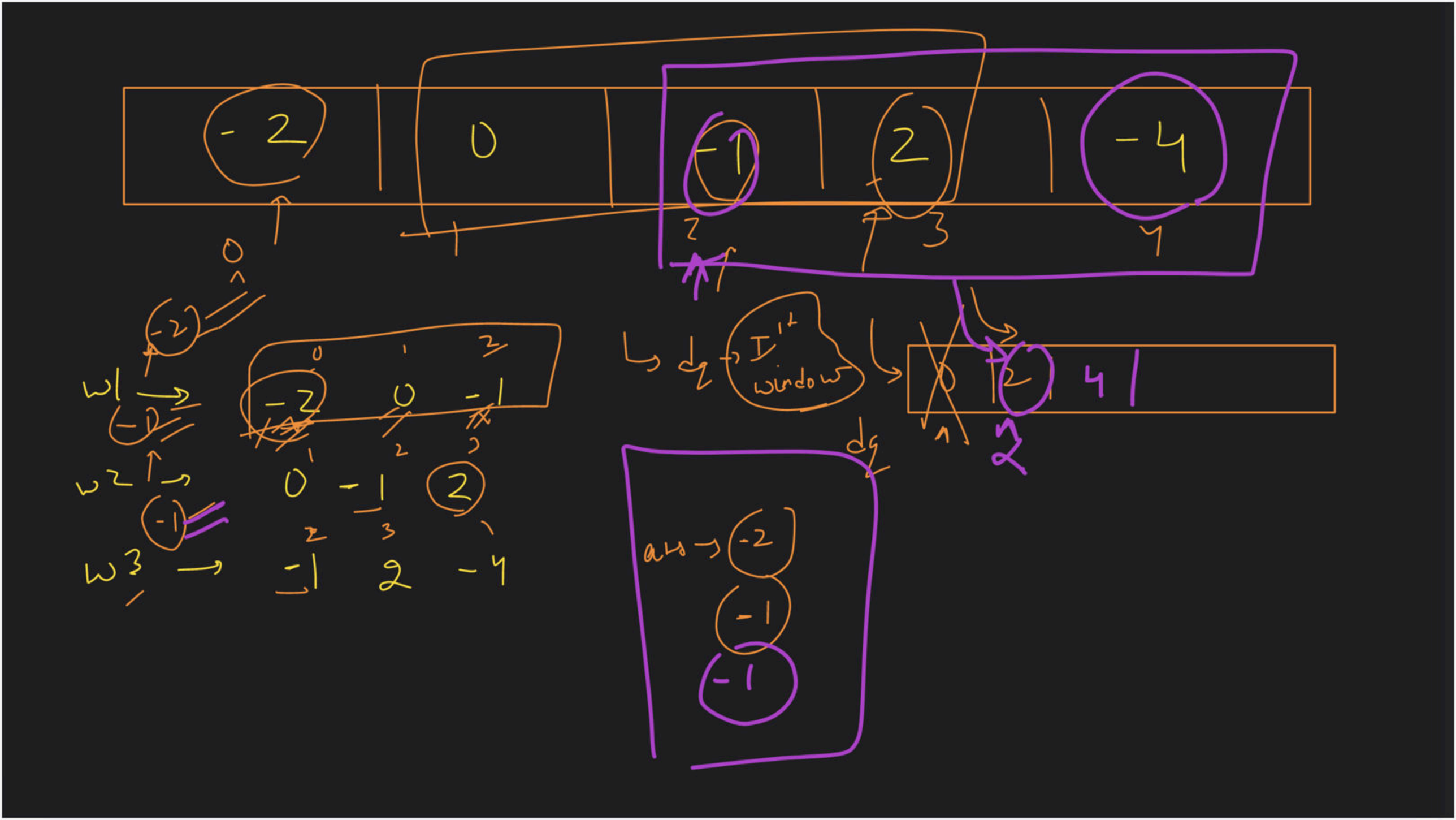
index - q.front >= K

3 - 0 >= 3

index - q.fr == K

index - q.fr < K


index
-2
0
4
-1
-5
6
arr
0
1
2
3
4
5
old
dq.front
w1
w1
index - dq.front >= K
out of range
K = 3
pop
w1 →
-2
0
4
3-3=0
3-2=1
3-1=2
w2
Ran
0
4
-1
Add
1 2 3
3-0=3
3-(-1)=4
<=0 index
out of range


-2
0
-1
2
-4
0
1
2
3
4
0
1
2
-2
0
-1
w1
-1
w2
-1
w3
0
-1
2
-1
2
-4
I'll window
dq
ans
-2
-1
-1
2