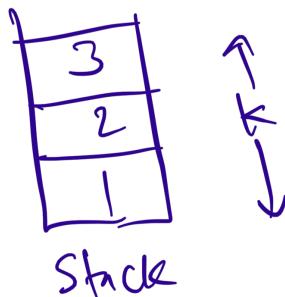
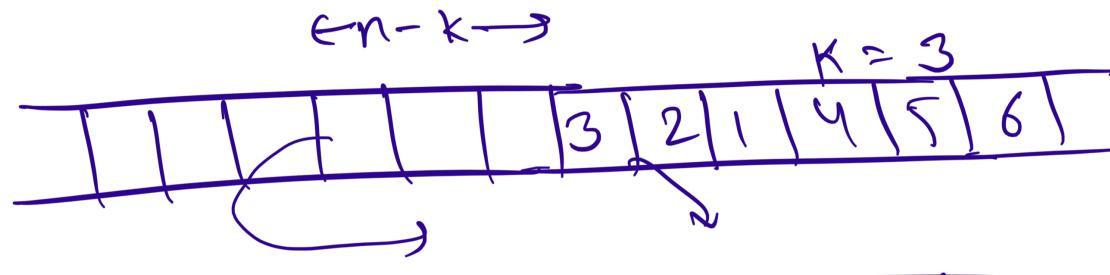
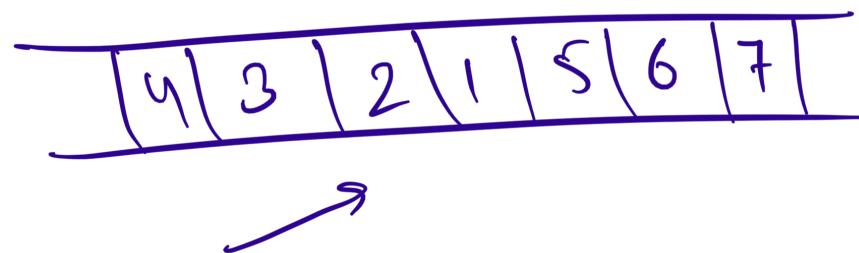
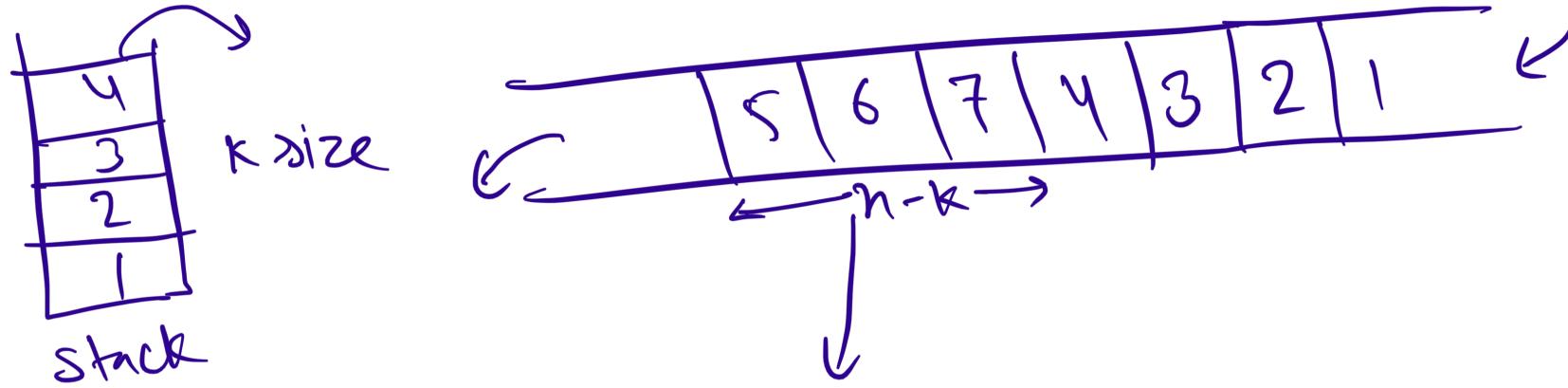
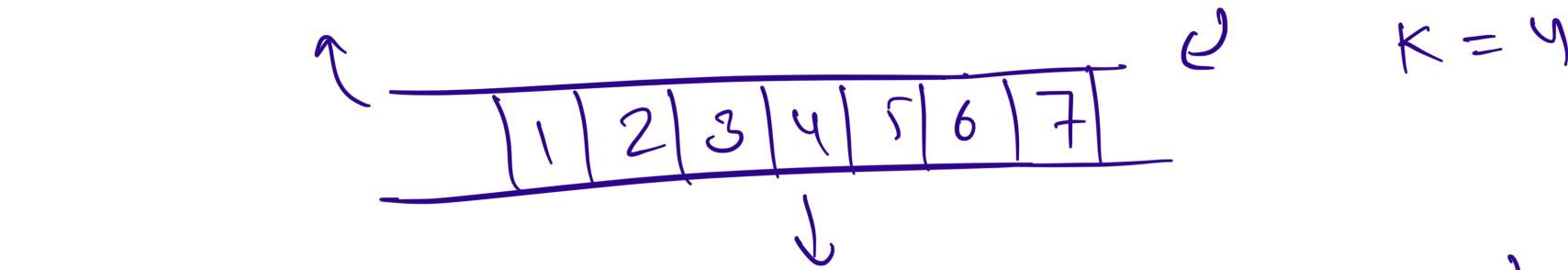


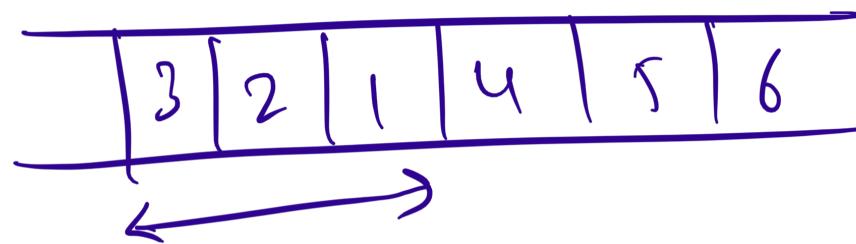
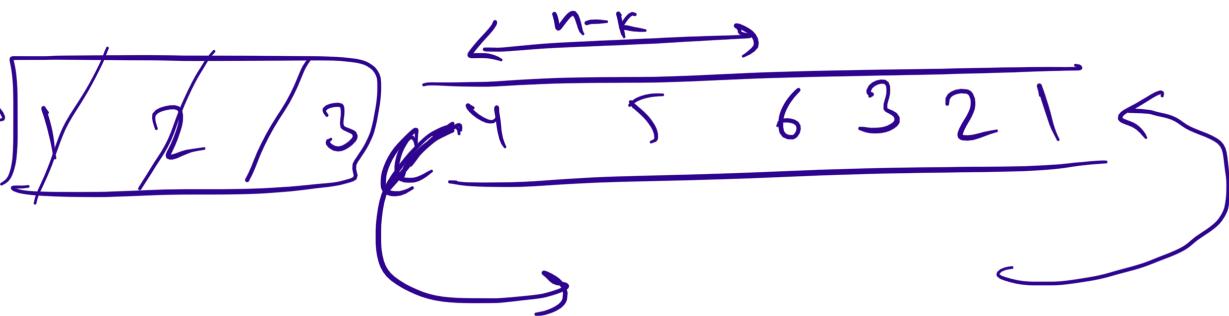
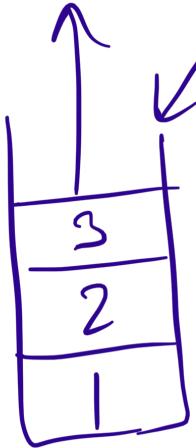
# Queue Problems - I

---

## Reverse the first k elements of a Queue



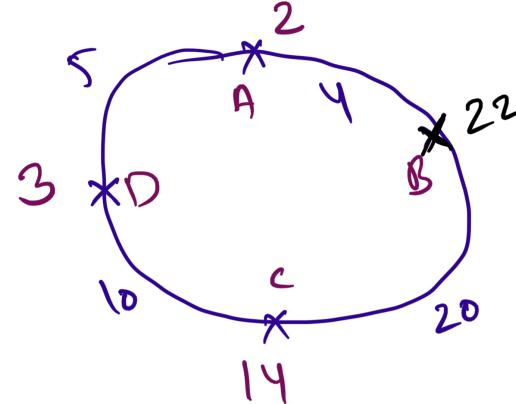




## Circular Tour Problem

Q. find the starting point such that one full circuit gets completed.

1 unit distance = 1 unit petrol



→ Tank capacity is unlimited

P → 

2	22	14	3
---	----	----	---

dis[] → 

4	20	10	5
---	----	----	---

$$22 - 20 = 2$$

$$2 + 14 = 16$$

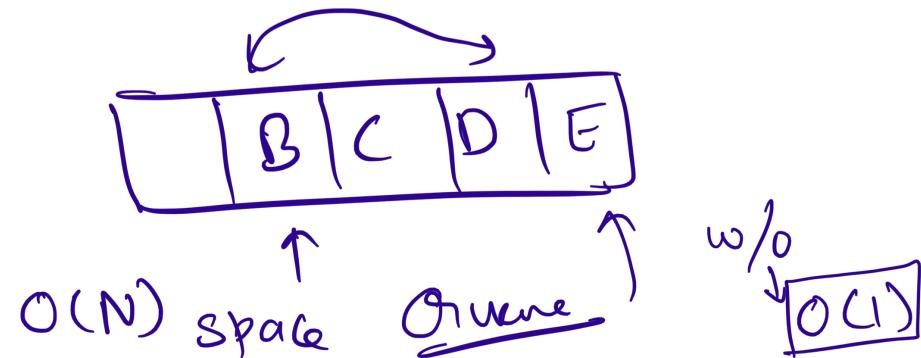
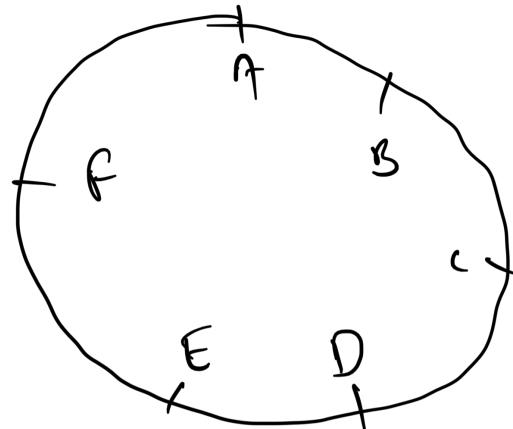
$$16 - 10 = 6$$

$$6 + 3 = 9$$

$$9 - 5 = 4$$

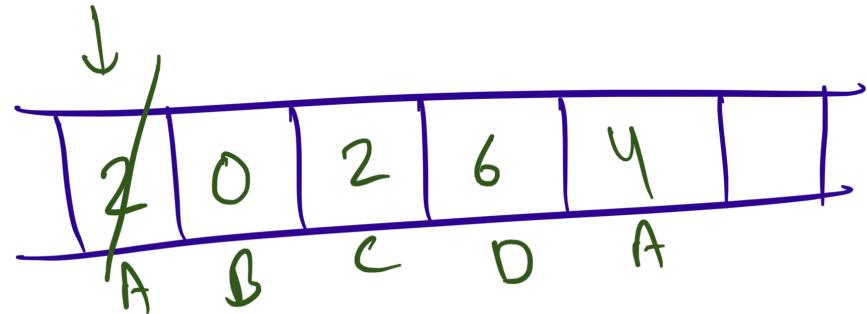
$\text{---} \quad O(N^2)$   

$$\begin{array}{r} A & B & C & D \\ \hline B & C & D & E & F \\ \times & & & & \\ C & D & & & \end{array}$$



$$4 + 2 = 6$$

$$6 - 4 = \boxed{2}$$



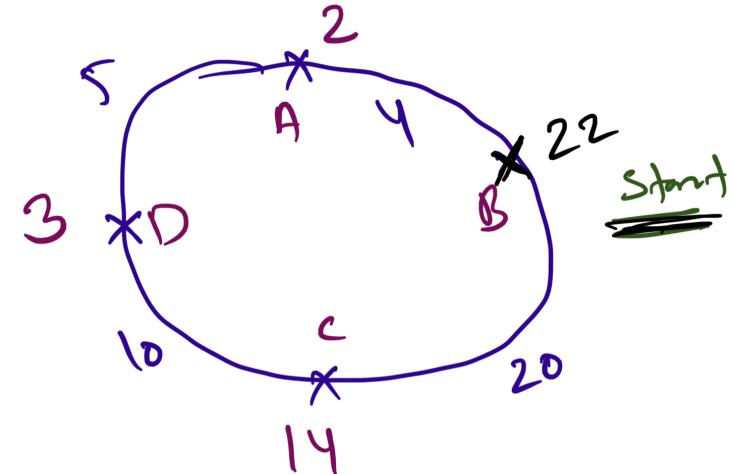
$$\text{curr} = 0 + 2 = 2$$

$$\begin{aligned}\rightarrow \text{curr} &= 2 - 4 = -2 - (2 - 4) \\ &= -2 - (-2) = 0\end{aligned}$$

$$\text{curr} = 0 + 22 = 22$$

$$= 22 - 20 = 2$$

$$\text{curr} = 2 + 14 = 16$$



$$\text{curr} = 16 - 10 = 6$$

$$6 + 3 = 9$$

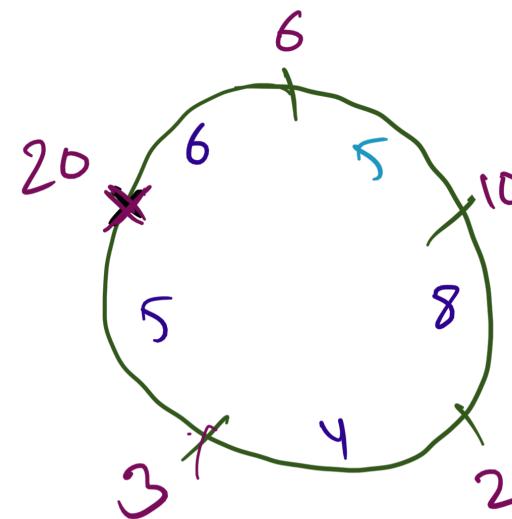
$$\text{curr} = 9 - 5 = 4$$

$$4 + 2 = 6$$

$$\text{curr} = 6 - 4 = 2 \checkmark$$

$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
6	10	2	3	20

$\leftarrow$	8	4	5	6
--------------	---	---	---	---



$$5 - 4 = 1$$

$$\text{curr} = 1 + 3 = 4$$

$$4 - 5 = -1$$

Start = 0  
 $w\gamma = \emptyset^6$   
 $\text{curr} = 6 - 5 = 1$   
 $1 + 10 = 11$   
 $\text{curr} = 11 - 8 = 3$   
 $3 + 2 = \underline{\underline{5}}$

$$cur = -1 - (6 - 5)$$

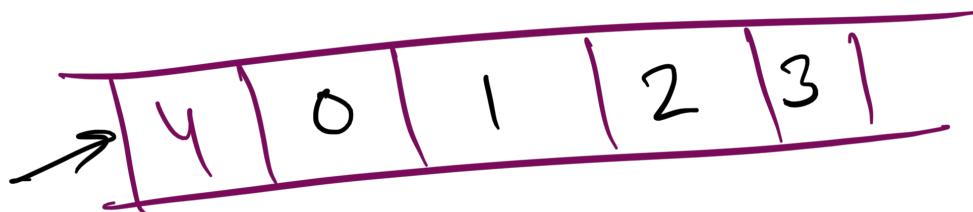
$$cur = -2 - (10 - 8)$$

$$= -2 - 2 = -4$$

$$= -4 - (2 - 4) = -4 - (-2) = -2$$

$$cur = -2 - (3 - 5) = -2 - (-2) = \underline{\underline{0}}$$

start = 4



index++

(index + 1) % n

$$curr = 0 + 20 = 20$$

$$20 - 6 = 14$$

$$curr = 14 + 6 = 20$$

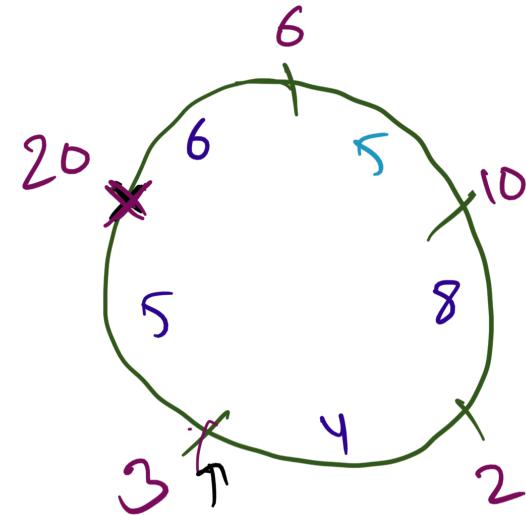
$$20 - 5 = 15$$

$$15 + 10 = 25$$

$$curr = 25 - 8 = 17$$

$$17 + 2 = 19 - 4 = 15 + 3 = 18$$

$$= 18 - 5 = \underline{\underline{13}} > 0$$



## Method -2

start = 0

$$\text{cur} = 0$$

prev = -1

$$\text{cur} = 0 + 6 - 5 = 1$$

$$\text{cur} = 1 + 10 - 8 = 3$$

$$\text{cur} = 3 + 2 - 4 = 1$$

$$\text{cur} = 1 + 3 - 5 = -1$$

[ cur = 0, start = 4

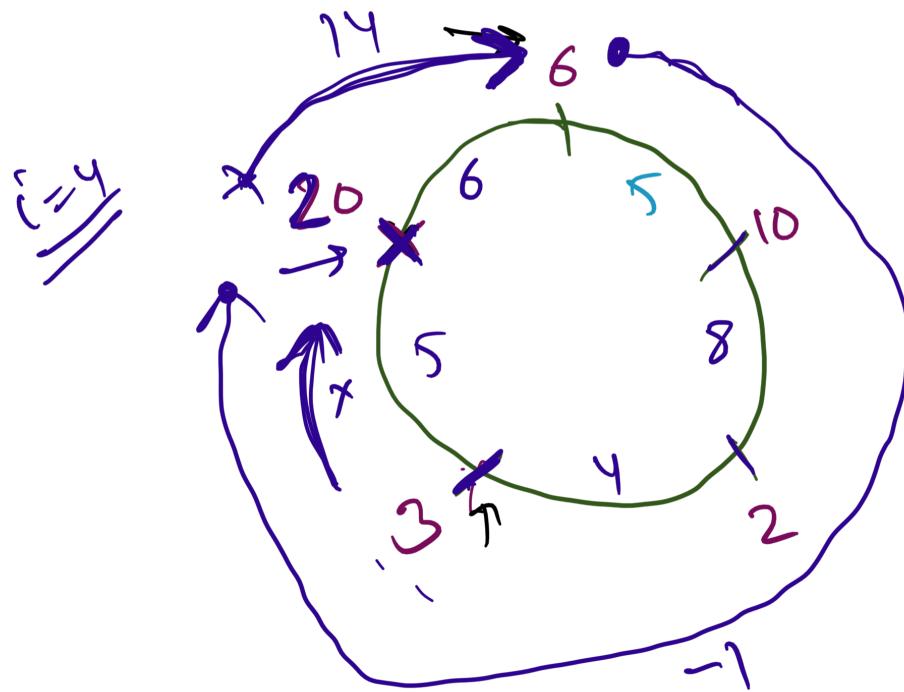
$$\text{cur} = 0 + 20 - 6 = \underline{\underline{14}}$$

10(N)

$$14 > \text{obj}(-1)$$

i=0	↓	↓	↓	↓	↓
P →	6	10	2	3	20

dis →	5	8	4	5	6
-------	---	---	---	---	---



$$\text{curr} + \text{prev} \geq 0$$

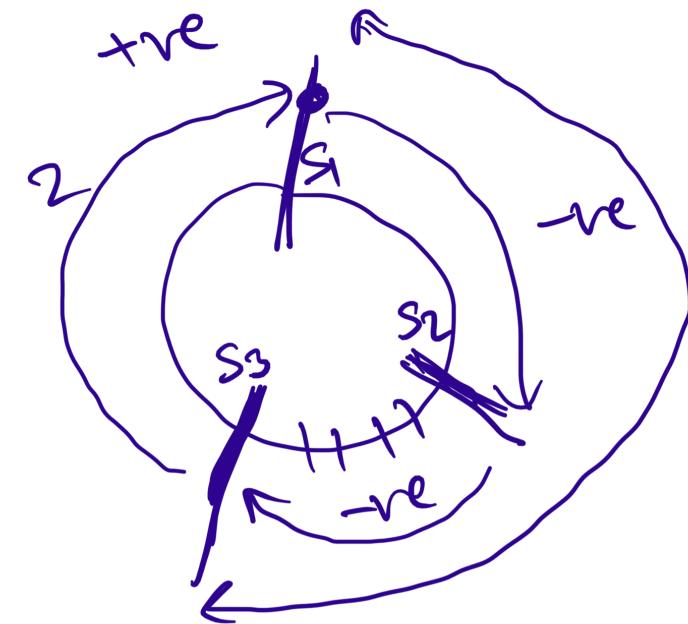
$$13 + (-1) = 12 \geq 0$$

$$\text{prev} = -5$$

$$\text{curr} = 2$$

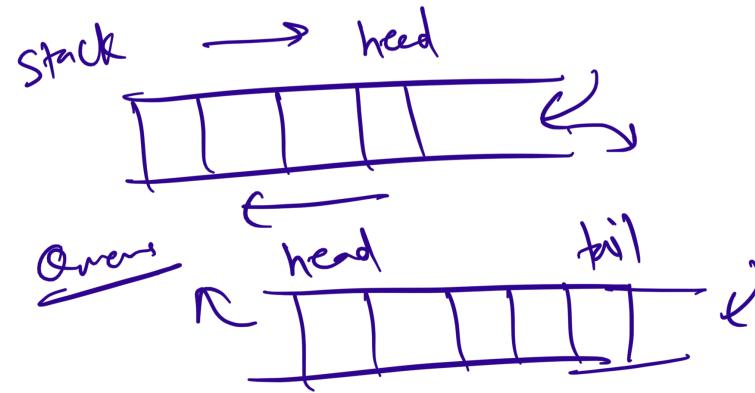
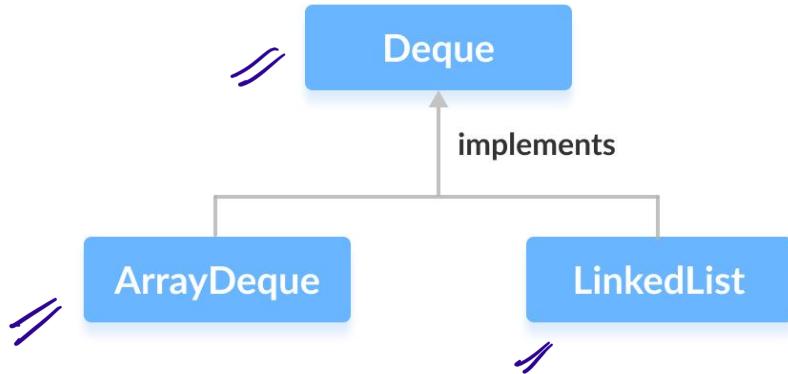
$$2 + (-5) = -3 < 0$$

(-1)



$$\text{prev} + =$$

# Deque Data Structure



## Methods of Deque

- **addFirst()** - Adds the specified element at the beginning of the deque. Throws an exception if the deque is full.
- **addLast()** - Adds the specified element at the end of the deque. Throws an exception if the deque is full.
- **offerFirst()** - Adds the specified element at the beginning of the deque. Returns `false` if the deque is full.
- **offerLast()** - Adds the specified element at the end of the deque. Returns `false` if the deque is full.
- **getFirst()** - Returns the first element of the deque. Throws an exception if the deque is empty.
- **getLast()** - Returns the last element of the deque. Throws an exception if the deque is empty.
- **peekFirst()** - Returns the first element of the deque. Returns `null` if the deque is empty.
- **peekLast()** - Returns the last element of the deque. Returns `null` if the deque is empty.
- **removeFirst()** - Returns and removes the first element of the deque. Throws an exception if the deque is empty.
- **removeLast()** - Returns and removes the last element of the deque. Throws an exception if the deque is empty.
- **pollFirst()** - Returns and removes the first element of the deque. Returns `null` if the deque is empty.
- **pollLast()** - Returns and removes the last element of the deque. Returns `null` if the deque is empty.

## Sliding Window Maximum Problem

$a[] = \{1, 3, 5, 2, 1, 4, 5, 3, 1, 2, 4\}$

$\rightarrow k=3$

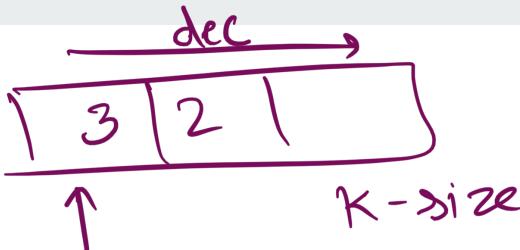
$ans[] = \{5, 5, 5, 4, 5, 5, 5, 3, 4\}$

$n-k+1$

$O(n*k)$

H.W.

Queue



Useful elements

\* head will always be the largest in current window

$a[] = \{1, 3, 5, 2, 1, 4, 5, 3, 5, 5, 4\}$



5, 5, 5, 4, 5, 5, 5, 5

T0(N) TC  
T0(K) SC

---

# Practice Problems

1. [Flatten a Multi-level Linked List](#)
2. Read more on Usage of Queue:  
<https://medium.com/tchie-delight/queue-data-structure-practice-problems-and-interview-questions-f459bf0578db>
3. Solve Problems on Stacks and Queues:  
<https://www.interviewbit.com/courses/programming/stacks-and-queues>