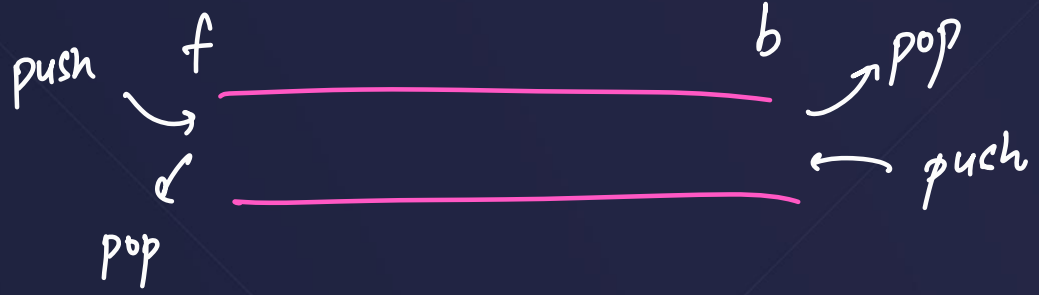
# Ques: Sliding Window Maximum [Leetcode - 239]




pop_front()
pop_back() ✓
push_front() ✓
push_back()

Monotonic Deque

↓

Decreasing

# Ques: Sliding Window Maximum [Leetcode - 239]

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| nums = { | 1, | 3, | -1, | -3, | 5, | 3, | 6, | 7 } |
| | | | | | | | | i |

k = 3

Window ke last me aane ke baad ans me push

f b

dq 7

ans = { 3, 3, 5, 5, 6, 7 }

```
while (nums[i] > dq.back()) pop_back;
    dq.push(nums[i])
if(i >= k-1)  ans.push(dq.front).
```

# Ques: Sliding Window Maximum [Leetcode - 239]

| | 0 | 1 | 2 |
|---|---|---|---|
| nums | 7 | 2 | 4 |
| | | j | i |

dq: 2

$K = 2$

$i - j + 1 = K$

$j = i - K + 1$

ans = {7, 4}

```
while (nums[i] > nums[dq.back]) dq.pop_back();
    dq.push_back(i);
while ( dq.front() < j ) dq.pop_front();
  ans.push_back( nums[dq.front] );
```

# Ques: Dota2 Senate

[Leetcode - 649]

2 parties Radiant & Dire

'R' 'D'

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| str | X | X | X | X | X | D | X |

q<int> q

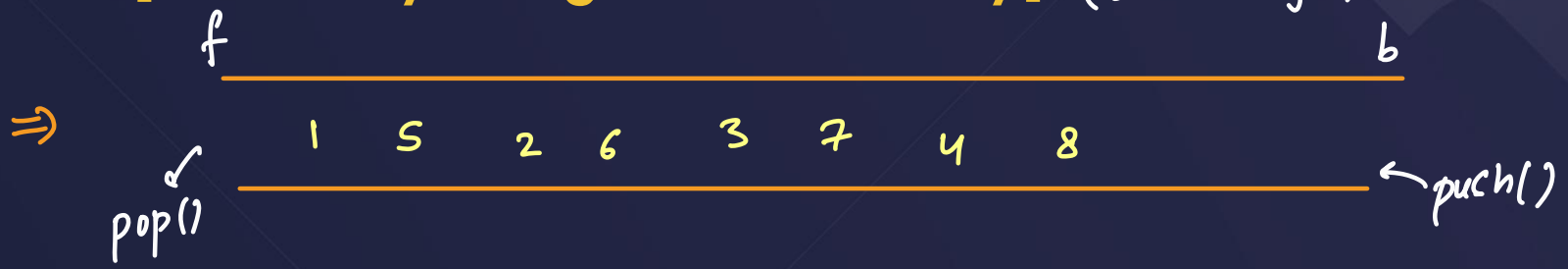

q<int> r

q<int> d


```
while(q.size()> 1){
```

# Ques: Reorder Queue (Interleave 1st Half with 2nd Half)

## [Do this by using one Stack only] (Even Length)




**Steps:**

1) First half ko q se st & then st se q

2) 1st (2nd) half q se st one by one, st se q, q se q

1 5 2 6 3 7 4 8

3) reverse the q

# Ques: Reveal Cards in Increasing Order [Leetcode - 950]

→ Show one, put the 2nd in back

```
Input: deck = [17,13,11,2,3,5,7]
Output: [2,13,3,11,5,17,7]
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| deck = | 2 | 3 | 5 | 7 | 11 | 13 | 17 |
| q = | | | | | | 5 | |
| ans = | 2 | 13 | 3 | 11 | 5 | 17 | 7 |

Done

```
int idx = q.front()
ans[idx] = deck[i]        (i = 0 to 6)
```

# Hint :
- Sort the array
- Use an extra array for result
- Use a queue

# THANK YOU!