Stacks -3

Lecture-47

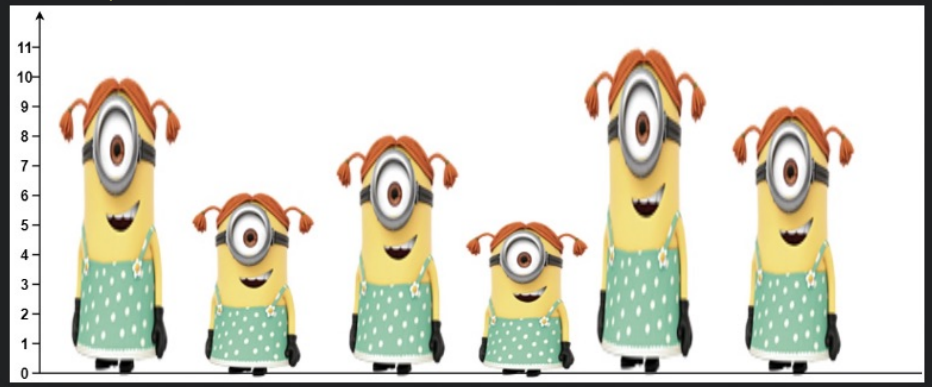
Raghav Garg

Today's checklist

1) Questions on Stacks

Ques: Number of Visible People in a Queue [Leetcode - 1944]

Example 1:



10 6 8 5 11 9

Input: heights = [10,6,8,5,11,9]

3 1 2 1 1 0

11 Count = 0 1

~~5 Count = 0 1~~

8 Count = 0 1 2

6 Count = 0 1

10: Count = 0 2 3

pop, ans, push

Ques: Sliding Window Maximum [Leetcode - 239]

M-I Brute Force

arr = { 1, 3, -1, -3, 5, 3, 6, 7 } k = 3 (i)

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

No. of window = n - k + 1

T.n.O = (n-k+1) * k

T.C. = O(n*k)

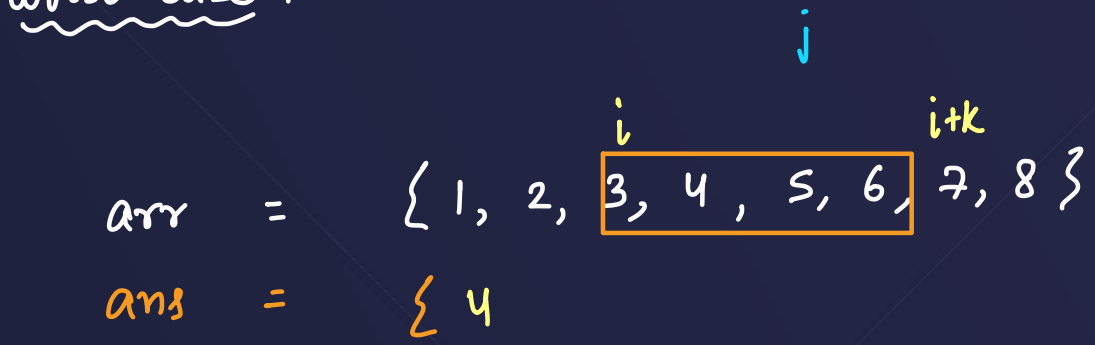
Ques: Sliding Window Maximum [Leetcode - 239]

Method-2 : Using a Stack [Next greater Element]

	0	1	2	3	4	5	6	7	8	
arr =	1,	3,	-1,	-3,	5,	3,	6,	7		k = 3
ngi =	1,	4,	4,	4,	6,	6,	7,	8		
ans =	3,	3,	5,	5,	6,	7				

Ques: Sliding Window Maximum [Leetcode - 239]

worst case :



O(n) T.C

K = 4

```
vector<int> ans;
for(int i=0;i<n-k+1;i++){
    int mx = arr[i]; // starting of window se
    int j = i;
    while(j < i+k){ // means if nge is inside window
        mx = arr[j];
        j = ngi[j];
    }
    ans.push_back(mx);
}
return ans;
```

mx = 1 1 2 3 4 2 7 4 5

Ques: Sliding Window Maximum [Leetcode - 239]

```
vector<int> ans;
int j = 0;
for(int i=0;i<n-k+1;i++){
    int mx = arr[j]; // starting of window se
    while(j < i+k){ // means if nge is inside window
        mx = arr[j];
        if(ngi[j] >= i+k) break;
        j = ngi[j];
    }
    ans.push_back(mx);
}
return ans;
```

if (j<i) ⇒ j=i

j

arr { 7, 2, 4 } k = 2

i i+K

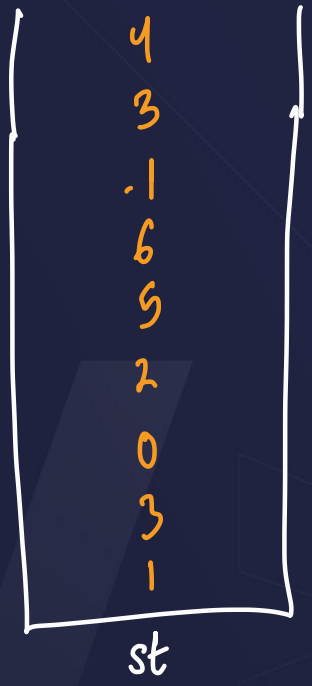
ans = { 7,

mx = 7 ~~2~~ 7

Ques: Min Stack

[Leetcode - 155]

Method-1 : Brute Force → T.C. = O(n)
S.C. = O(n)

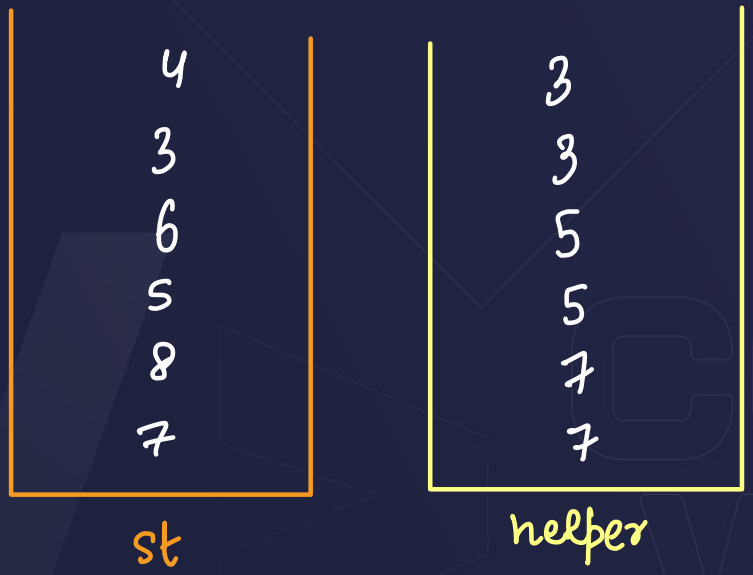


min = 6 5 2 0

Ques: Min Stack [Leetcode - 155]

Method-2 : By using extra stack that will always have the same no. of elements as your given st.

T.C. = O(1) , S.C. = O(n)




```
push(val){
    st.push(val);
if (val < helper.top())
    helper.push(val);
else (val >= helper.top()
    helper.push(helper.top());
```

Ques: Min Stack

[Leetcode - 155]

Method-3 S.C. = O(1) , T.C. = O(n)

Implement the stack with vector

& each time you call getmin → O(n)

you have to traverse the entire vector.

Ques: Min Stack [Leetcode - 155]

Method - 4 : T.C. = O(1) , S.C. = O(1)

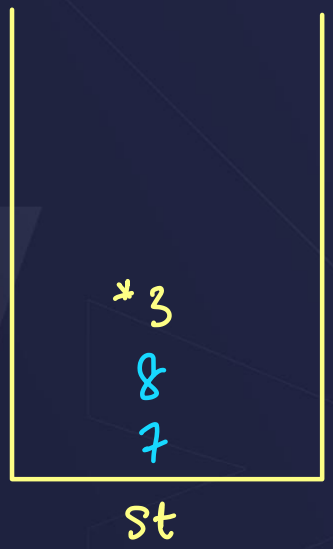
Tagdi math

min = ~~7~~ ~~5~~ ~~3~~ 5

getmin() → 3
pop()
pop()
pop()
pop()

```
push(val) {
    if (val < min) {
        st.push(val + (val - min))
        min = val
    }
}
```

3



5 + (5 - 7)

old = 2*3 - 1 = 6 - 1 = 5

2*min - oldmin = st.top()

oldmin = 2*min - st.top()

min = oldmin

$\Rightarrow \quad val < min$

$\Rightarrow \quad val - min < 0$

$\Rightarrow \quad \boxed{val + (val - min) < val}$

THANK YOU!