

My Observation.

0	2	3	1	1	1	0	2	3	1	0	1	1	0	2	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First thought was that just counting contiguous segments having same value shall give me the answer.

Meaning?

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	3	1	1	1	0	2	3	1	0	1	1	0	2	3

So for deleting 1's

group 1

group 2

group 3

3 deleting steps to make all 1's → 0's

or rather making where would this go wrong

then  $\{i=2, j=12\} \Rightarrow$  if there were no 0's in b/w then we could have made 0 in 1 step.

My Observation.

0	2	3	1	1	1	0	2	3	1	0	1	1	0	2	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

So I observed a pattern,

Let's say the array is like

2 2 2 2 3 3 3 3 4 4 4 (1 1) 4 4 4

Notice this, as soon as a number smaller than the previous number appears we will have to break the space.

For example.

(2 2 2) (3 3 3) (4 4 4)  
group1 group2 group3

these 3 groups can be separately made into '0' one by one



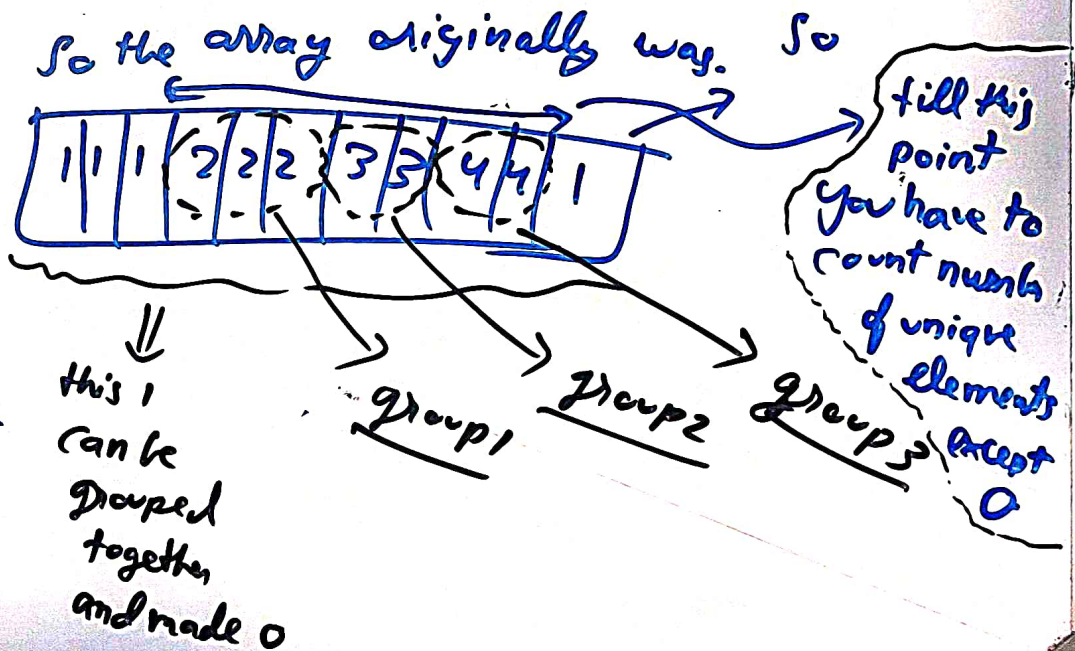
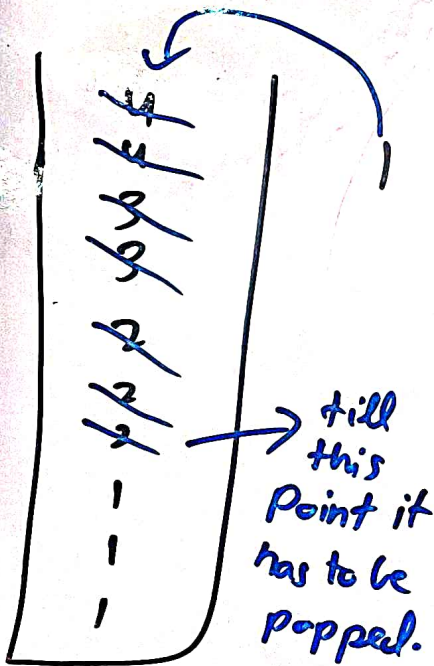
My Observation.

2 2 2    3 3 3    4 4 4 4    1 1 1 .....  
↑↑  
break the group here

So as soon as a number

$\text{nums}[i] < \text{stack.top}()$

→ those numbers have to be popped



$op = 0$   
 for ( $i = 0 \dots N-1$ ) {  
     Set

    while ( $!st.empty() \ \& \ st.top() > num[i]$ ) {  
         Set.add(st.pop());



    }  
      $op += st.size()$  → except '0'  
     st.push(num[i])

}

    Set  
     while ( $!st.empty()$ ) {  
         Set.add(st.pop());  
     }  
     }

For the case

1 1 2 2 3 3 4 4

$op += Set.size()$