

## 41. First Missing Positive

arr:

15	8	27	3	1	6	9	2
0	1	2	3	4	5	6	7

$n$  : arr. length

ans: 1 to  $n+1$

-15	-8	-27	3	1	-6	9	-2
0	1	2	3	4	5	6	7
↓	↓	↓	↓	↓	↓	↓	↓
1	2	3	4	5	6	7	8

$T : O(n)$

$S : O(1)$

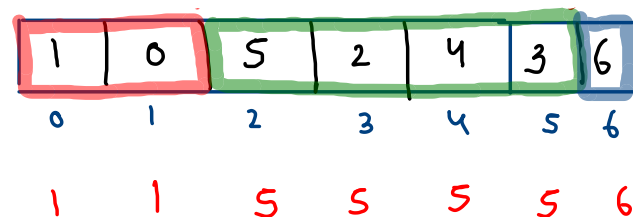
## 769. Max Chunks To Make Sorted

You are given an integer array `arr` of length `n` that represents a permutation of the integers in the range `[0, n - 1]`.

We split `arr` into some number of **chunks** (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array.

Return *the largest number of chunks we can make to sort the array*.

$$n = 5$$



2	0	1	5	3	6	4	7	8
0	1	2	3	4	5	6	7	8

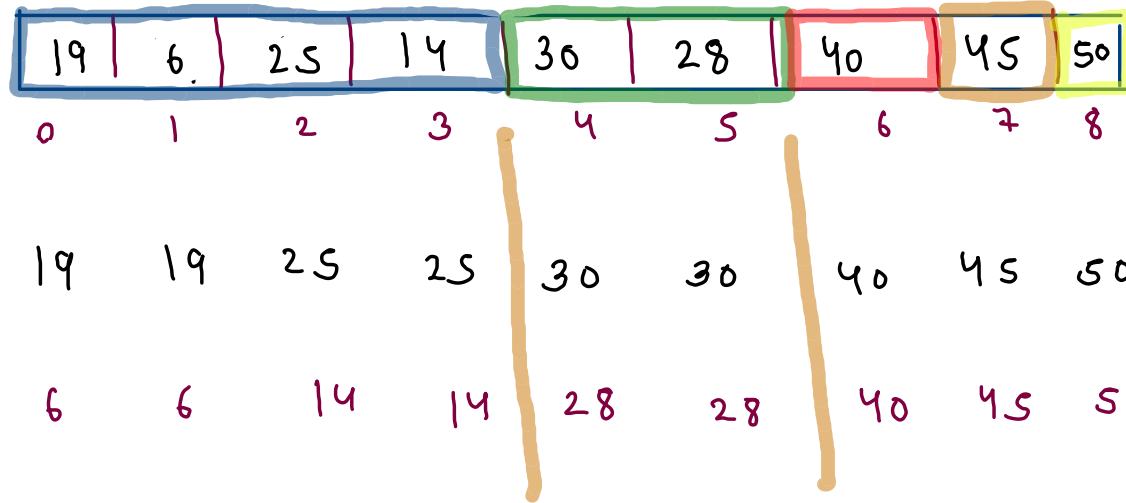
max

2 2 2 5 5 6 6 7 8

## 768. Max Chunks To Make Sorted II

S: extra space

T:  $O(n)$



## 628. Maximum Product of Three Numbers

5	-9	3	-8	6	1
0	1	2	3	4	5

$$j_{\max} = 6$$

$$s_{\max} = 5$$

$$t_{\max} = 3$$

$$j_{\min} = -9$$

$$s_{\min} = -8$$

$$c1 = j_{\max} * s_{\max} * t_{\max}$$

$$c2 = j_{\min} * s_{\min} * j_{\max}$$

$$\text{return } \max(c1, c2)$$

