Week 7 LIVE

Array Problems And Doubts Session

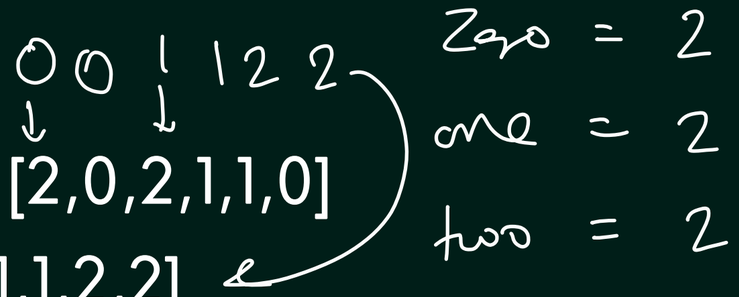
In This Lecture

1. Sort Colors
2. Max Chunks To Make Sorted - I
3. Max Chunks To Make Sorted - II

Sort Colors

Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively.



Input: nums = [2,0,2,1,1,0]

Output: [0,0,1,1,2,2]

Sort Colors

zeroes ← l

twos h →

a[m] == 2 ⟶ a[m] = a[h]
a[h] = 2; h--;

a[m] == 0 ⟶ a[m] = a[l]
a[l] = 0
l++; m++;

a[m] == 1 ⟶ m++;

Sort Colors

zeroes ← l

twos h →

a[m] == 2 ⟶

```
a[m] = a[h]
a[h] = 2; h--;
```

a[m] == 0 ⟶

```
a[m] = a[l]
a[l] = 0
l++; m++;
```

a[m] == 1 ⟶

```
m++;
```

Max Chunks To Make Sorted - I

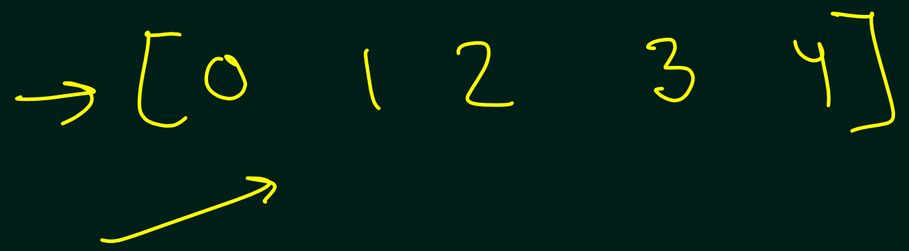
You are given an integer array A of length n that represents a permutation of the integers in the range [0, n - 1].

We split A into some number of Partitions, and individually sort each chunk. After joining them, the result should equal the sorted array.

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

Input:

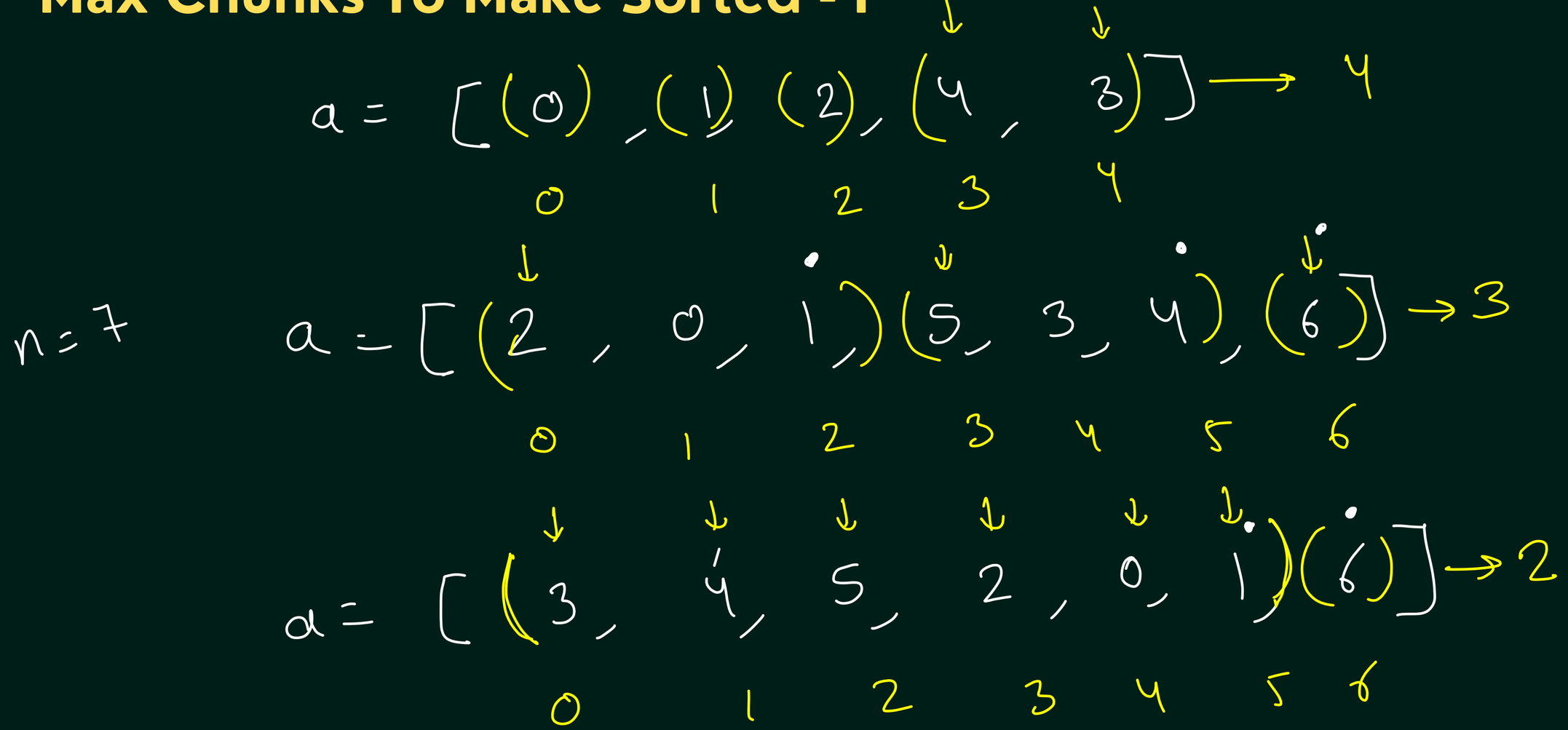
[0, 1, 2, 4, 3]



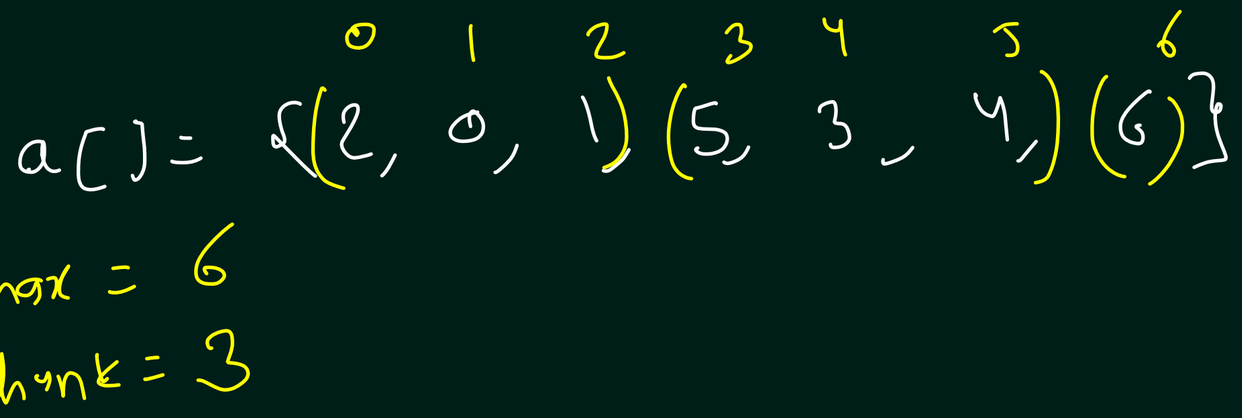
Output:

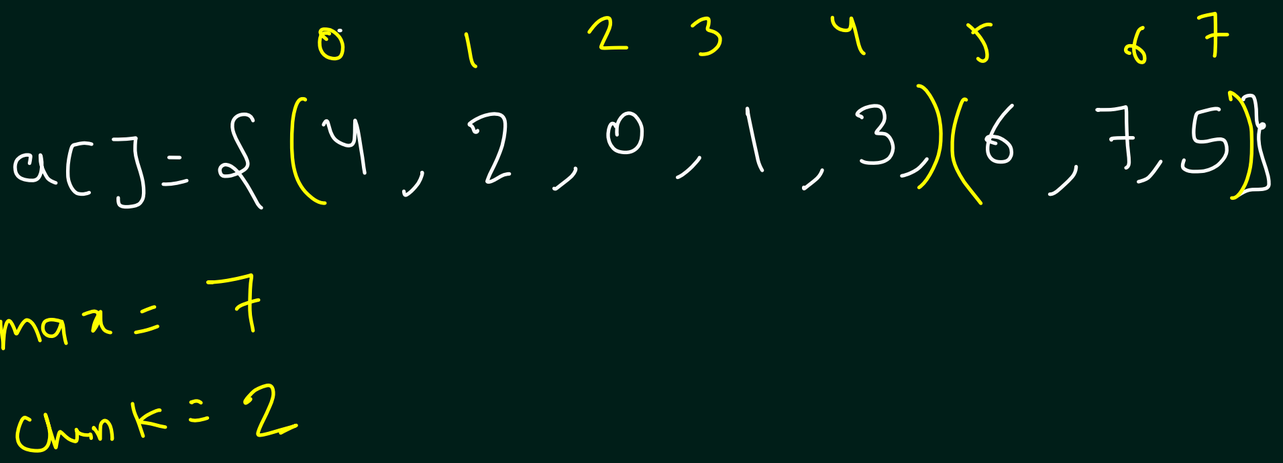
4

Max Chunks To Make Sorted - I



Max Chunks To Make Sorted - I






```java
static int maxChunk1(int a[]) {
    if(a.length == 0) return 0;
    int chunks = 0;
    int maxSoFar = a[0];

    for(int i = 0; i<a.length; i++) {
        maxSoFar = Math.max(maxSoFar, a[i]);
        if(i == maxSoFar) {
            chunks++;
        }
    }
    return chunks;
}
```

Max Chunks To Make Sorted - II

You are given an integer array arr.
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.

Input:
arr = [8, 2, 5, 2]
Output:
1

Max Chunks To Make Sorted - II

a[] = { (6, 3, 5) | (8, 7) | (12, 11, 9) }

6 | 7 | 8 | 9

(3, 6) → (7, 8) → (9, 12)

a[] = { 0, 1, 2, 3, 4 }

a[] = { [1, 0], [3, 2], [4] }

Max Chunks To Make Sorted - II