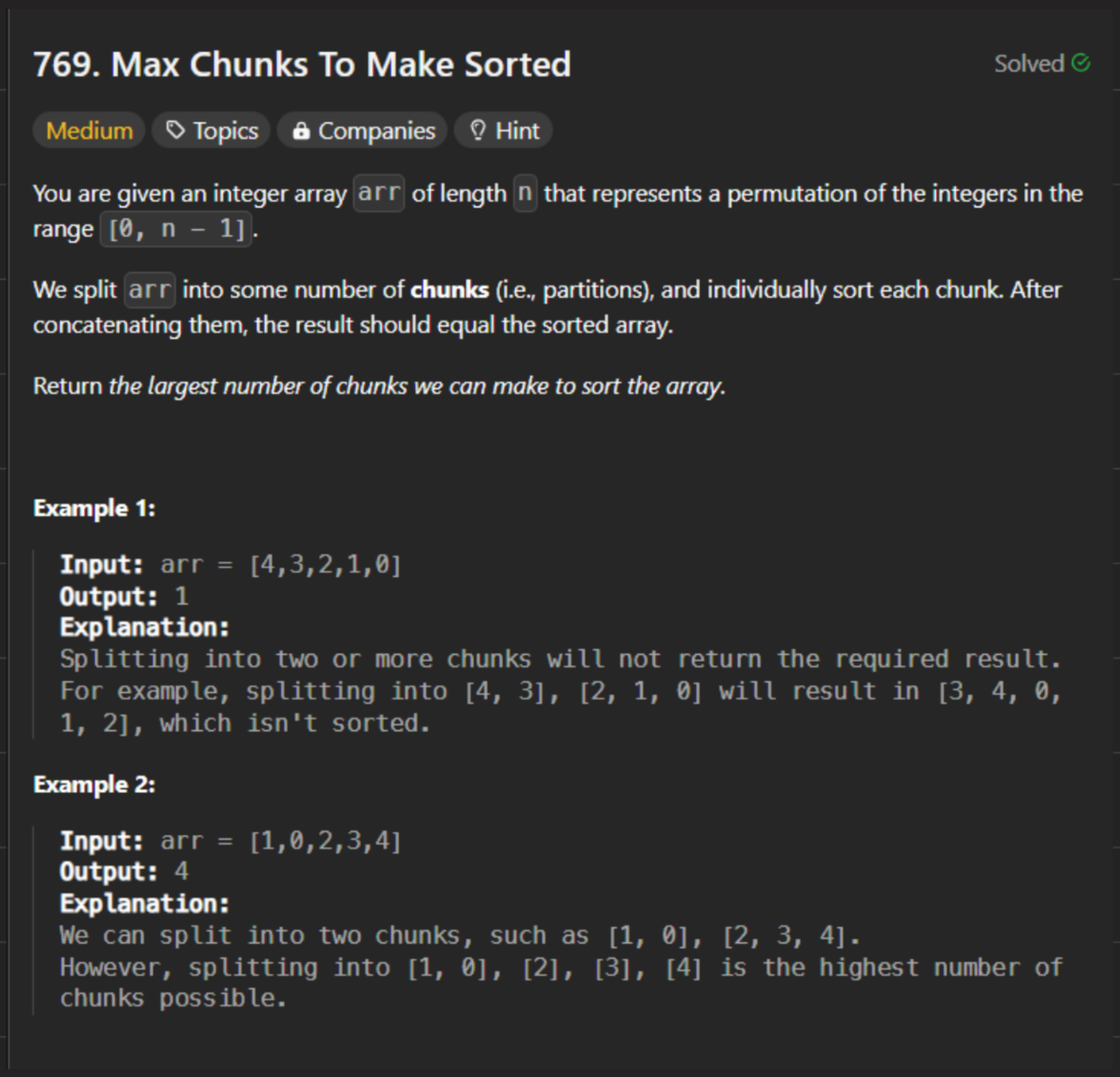
Max Chunks to make array Sorted -I



769. Max Chunks To Make Sorted

Solved ✓

Medium · Topics · Companies · Hint

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.*

Example 1:

```
Input: arr = [4,3,2,1,0]
Output: 1
Explanation:
Splitting into two or more chunks will not return the required result.
For example, splitting into [4, 3], [2, 1, 0] will result in [3, 4, 0,
1, 2], which isn't sorted.
```

Example 2:

```
Input: arr = [1,0,2,3,4]
Output: 4
Explanation:
We can split into two chunks, such as [1, 0], [2, 3, 4].
However, splitting into [1, 0], [2], [3], [4] is the highest number of
chunks possible.
```

Ex :- $(\overset{0}{1}, \overset{1}{0}, \overset{2}{2}, \overset{3}{3}, \overset{4}{4})$

The key understanding is, if we Sort the elements → The obtained answer will be Our indexes

$(\overset{0}{1}, \overset{1}{0}, \overset{2}{2}, \overset{3}{3}, \overset{4}{4}) \xrightarrow{\text{Sort}} (\overset{0}{0}, \overset{1}{1}, \overset{2}{2}, \overset{3}{3}, \overset{4}{4})$

indexes

values

* So, the question will be a permutation of indexes only, it won't go beyond range of indexes

So, what are chunks here ? → "partitions"

arr(5) : $(1, 0, \mid 2, \mid 3, \mid 4)$

↓

if we split the array into 4 parts and sort each part individually, the whole array will be sorted

∴ Ans is 4 chunks

for same example:

$(0, 1, \mid 2, 3, 4)$

↑

we can split as well, if you sort those 2 individual chunks then also we get a sorted whole array,

but, we need to maximize chunks

arr[9] : (2, 0, 1 | 4, 3 | 6, 7, 5 | 8)

↓ sort individually

(1, 0, 2 | 3, 4 | 5, 6, 7 | 8) ✓ Array is sorted

① ② ③ ④

Ans = 4 chunks

How ?

```
 0  1  2  3  4  5  6  7  8
(2, 0, 1, 4, 3, 6, 7, 5, 8)
```

* if you want to make chunks, look for maximum index of the Chunks

```
 0  1  2 | 3  4 | 5  6  7 | 8
(2, 0, 1 | 4, 3 | 6, 7, 5 | 8)
```

All indexes and values are bound to be in this chunk only

i	arr[i]	maxValue	
		~~-∞~~	
0	2	2	
1	0	2	
2	1	2	→ in this case, index & maxValue upto this point are equal, so make a chunk here
3	4	~~2~~ 4	
4	3	4	→ chunk again
5	6	~~4~~ 5	
6	7	~~5~~ 6	
7	5	~~6~~ 7	→ chunk again
8	8	~~7~~ 8	→ chunk again

* So, simply if you look for (maxValue & IndexPosition), you can make chunks

Note :- If already a number is placed in its perfect index, chunk++