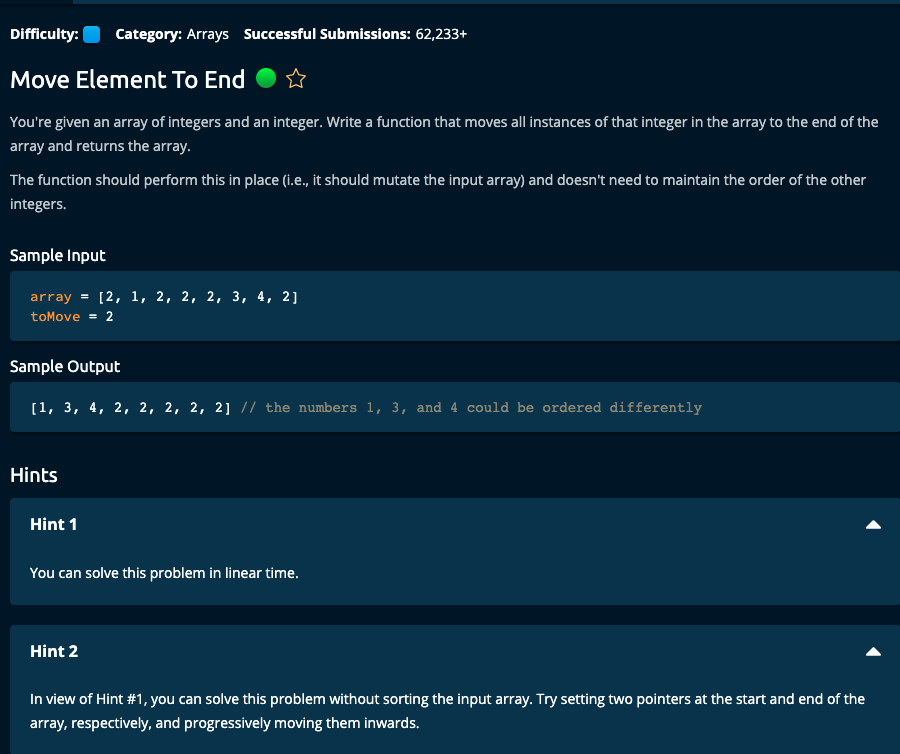
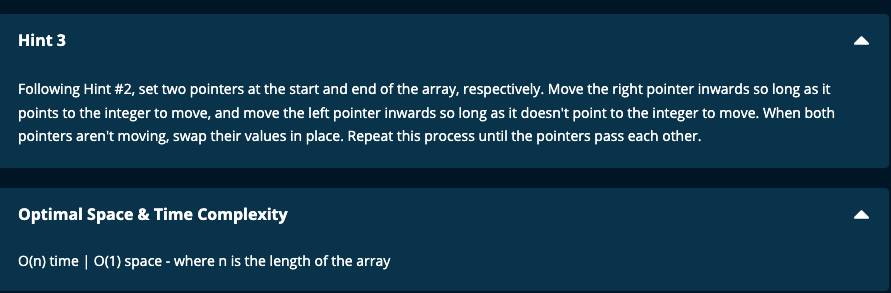
Move Element To End (Medium) <https://www.algoexpert.io/questions/move-element-to-end>





My Solutions:

Solution 1:

def moveElementToEnd(array, toMove):

n = len(array)

left = 0

right = n - 1

while left < right:

while array[left] != toMove and left < right:

left += 1

while array[right] == toMove and left < right:

right -= 1

if array[left] == toMove and array[right] != toMove:

array[left], array[right] = array[right], array[left]

left += 1

right -= 1

return array

-----------------------

Solution 2:

def moveElementToEnd(array, toMove):

n = len(array)

left = 0

right = n - 1

while left < right:

while array[right] == toMove and left < right:

right -= 1

if array[left] == toMove:

array[left], array[right] = array[right], array[left]

left += 1

return array

==========================================

JJ Notes:

1. Set the left pointer to 0 and right pointer to the end of the array.
2. Move the pointers towards each other until they cross.
3. First move the right pointer towards the left until we have an element is not equal to toMove and is a candidate for moving to the left. Check the array element at the left pointer. If its value is toMove, then swap the elements at the left and right pointer.

Then increment the left pointer.

1. Finally return the array.

Time Complexity = O(n) and Space complexity : O(1)

Algoexpert Solution:

def moveElementToEnd(array, toMove):

i = 0

j = len(array) - 1

while i < j:

while i < j and array[j] == toMove:

j -= 1

if array[i] == toMove:

array[i], array[j] = array[j], array[i] # Swap

i += 1 # Advance i

return array

Time Complexity = O(n) and Space complexity : O(1)