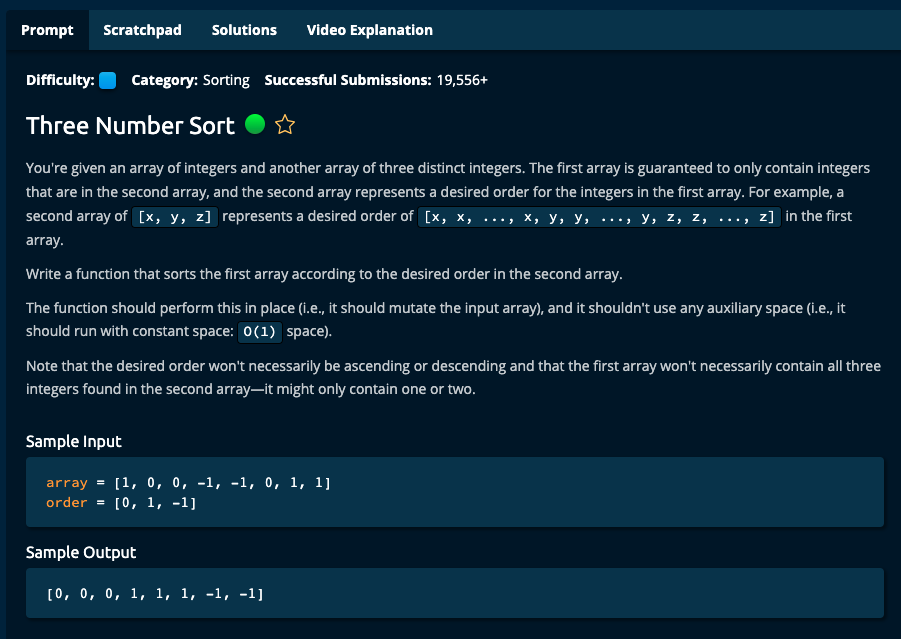
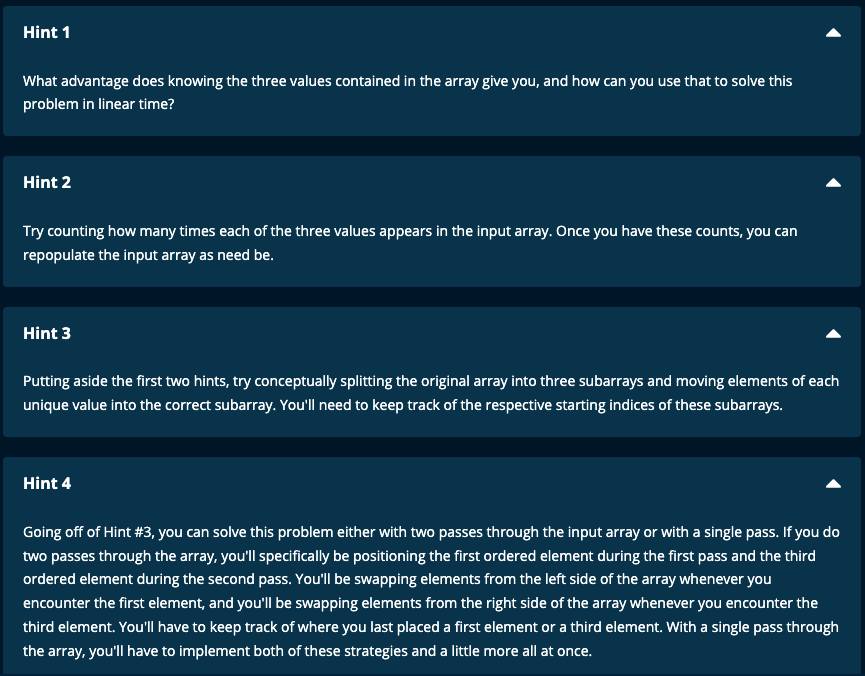
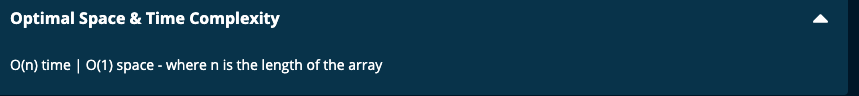
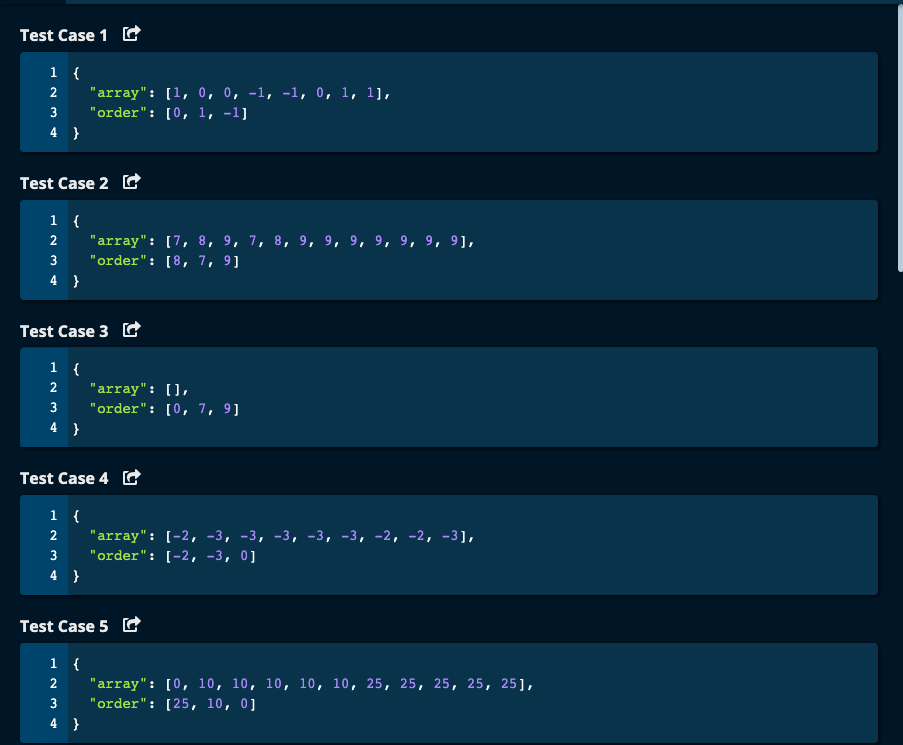
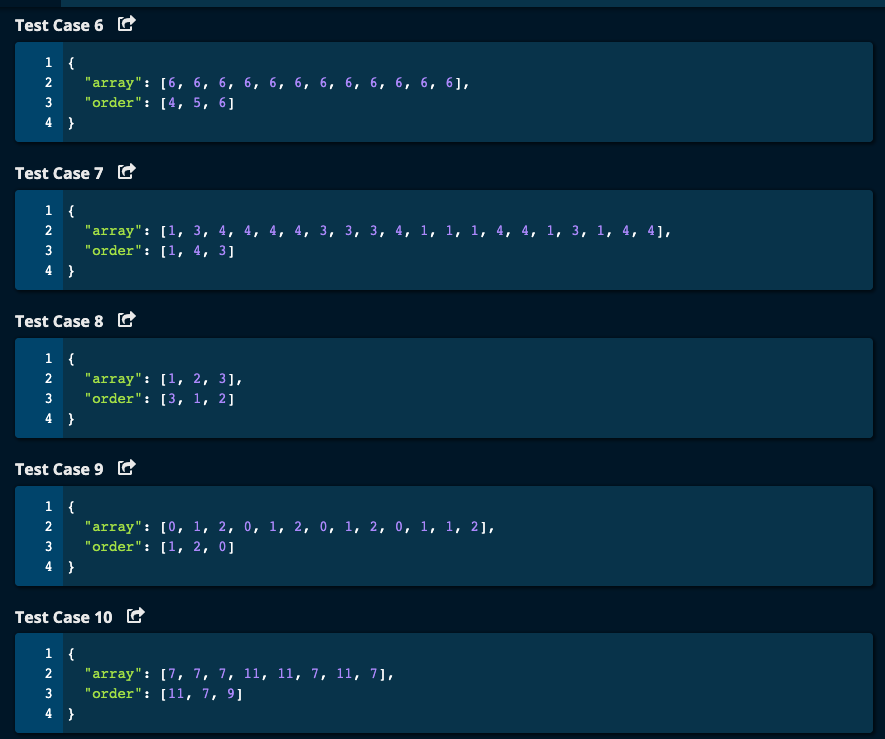
Three Number Sort. (Medium)

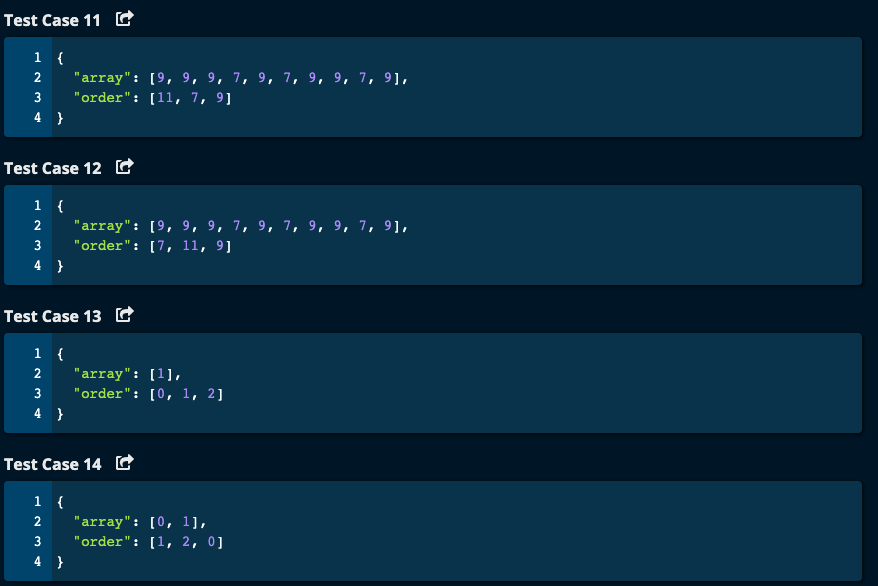












My Solution:

1. Get the number of occurrences of each of the 3 numbers in order using a dictionary as we traverse the array.
2. Repopulate the array with these numbers and each number occurs as many times as given by the values in the dictionary.
3. We make 2 passes through the array.
4. Time Complexity: O(n)

Space Complexity: O(1) for the dictionary.

def threeNumberSort(array, order):

mydict = {order[0]: 0, order[1]: 0, order[2]: 0}

for num in array:

if num == order[0]:

mydict[order[0]] += 1

elif num == order[1]:

mydict[order[1]] += 1

else:

mydict[order[2]] += 1

k = 0

for i in range(len(order)):

for j in range(mydict[order[i]]):

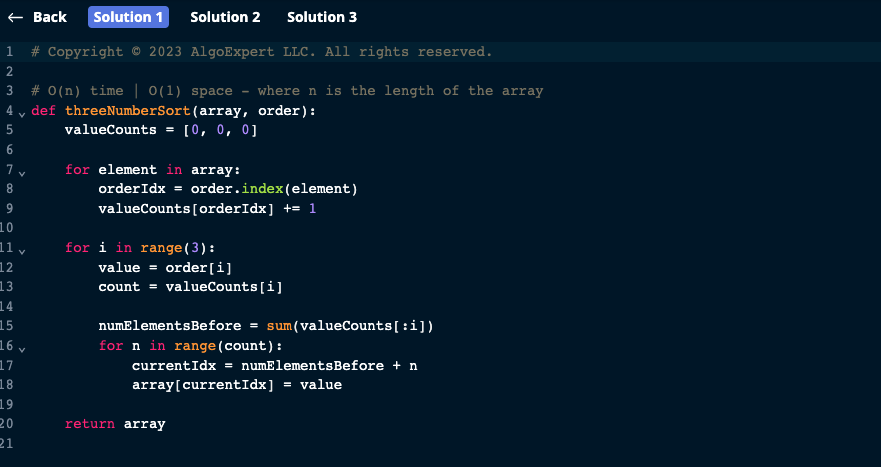
array[k] = order[i]

k += 1

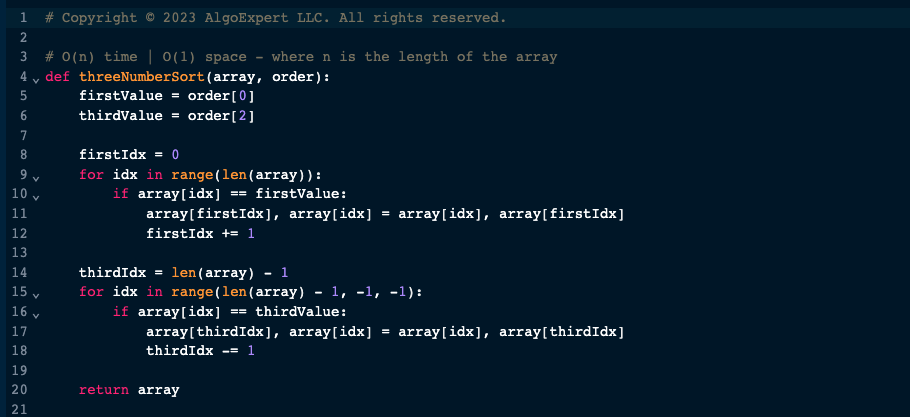
return array

Algoexpert Solutions:

Solution 1:



Solution 2:



Solution 3:

