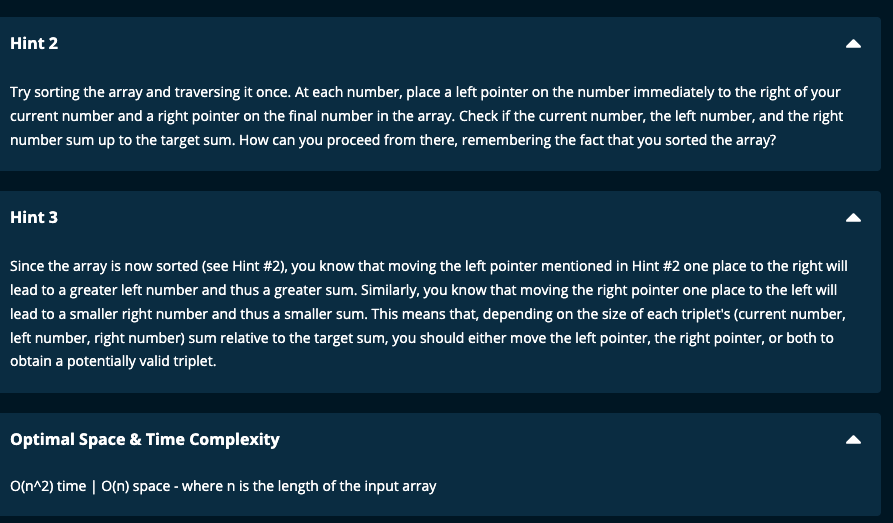
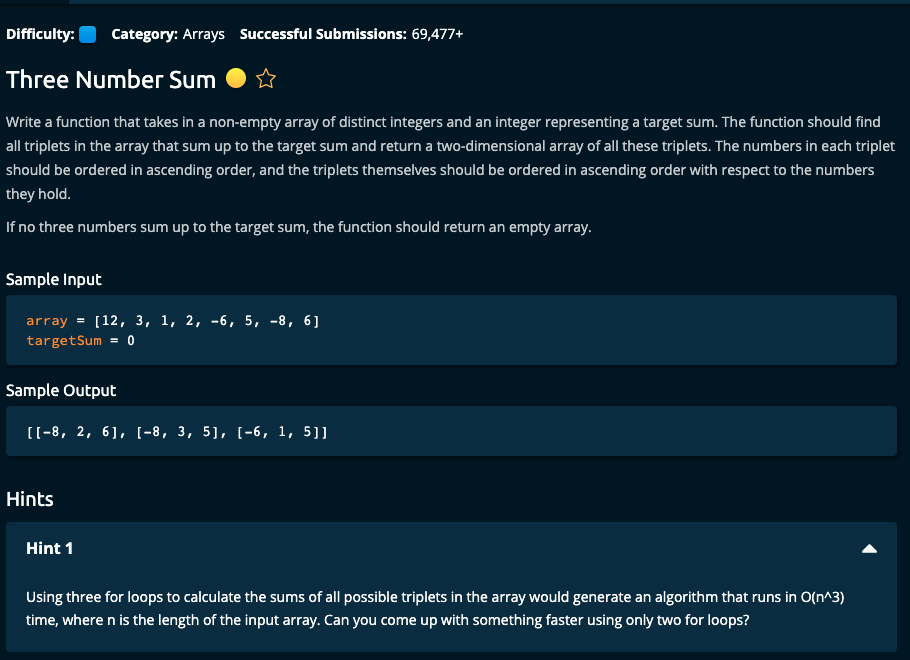
Three Number Sum. (Medium)

<https://www.algoexpert.io/questions/three-number-sum>



My Solution 1:

def threeNumberSum(array, targetSum):

array = sorted(array)

n = len(array)

result = []

n = len(array)

for i in range(n - 2):

target = targetSum - array[i]

left = i + 1

right = n - 1

while left < right:

if array[left] + array[right] == target:

result.append([array[i], array[left], array[right]])

left += 1

right -= 1

elif array[left] + array[right] >= target:

right -= 1

else:

left += 1

return result

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

My Solution 2:

def threeNumberSum(array, targetSum):

array = sorted(array)

result = []

for i in range(len(array) - 2):

target = targetSum - array[i]

res = twoSum(array, target, i)

result += res

return result

def twoSum(array, target, i):

left = i + 1

right = len(array) - 1

res = []

while left < right:

if array[left] + array[right] == target:

res.append([array[i], array[left], array[right]])

left += 1

right -= 1

elif array[left] + array[right] >= target:

right -= 1

else:

left += 1

return res

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

Notes:

1. Sort the array.
2. Have a for loop and run through the array except for the last two numbers.
3. Find out the target for the other two numbers (target = targetSum – array[i])
4. Have a left and right pointer and go through the array from left and right towards each other as in Two Sum.
5. In my second solution, I have made TwoSum into a separate function.
6. If the target is equal to (targetSum – array[i]) then advance both pointers towards each other after appending the triplet list to the result array. If the target is less, then move the left pointer to the right (i.e. increment left pointer), and if target is more, then move the right pointer to the left (i.e. decrement right pointer)
7. Time complexity: O(n^2) |. Space complexity = O(n)

Sorting in the array takes O(n.log(n)) which is less than O(n^2)

Algoexpert Solution:

def threeNumberSum(array, targetSum):

array.sort()

triplets = []

for i in range(len(array)- 2):

left = i + 1

right = len(array) - 1

while left < right:

currentSum = array[i] + array[left] + array[right]

if currentSum == targetSum:

triplets.append([array[i], array[left], array[right]])

left += 1

right -= 1

elif currentSum > targetSum:

right -= 1

elif currentSum < targetSum:

left += 1

return triplets