

Three Sum

15. 3Sum Solved

Medium Topics Companies Hint

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Notice that the solution set must not contain duplicate triplets.

Example 1:
Input: nums = [-1, 0, 1, 2, -1, -4]
Output: [(-1, -1, 2), (-1, 0, 1)]
Explanation:
nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.
nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.
nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.
The distinct triplets are [-1, 0, 1] and [-1, -1, 2].
Notice that the order of the output and the order of the triplets does not matter.

Example 2:
Input: nums = [0, 1, 1]
Output: []
Explanation: The only possible triplet does not sum up to 0.

Example 3:
Input: nums = [0, 0, 0]
Output: [(0, 0, 0)]
Explanation: The only possible triplet sums up to 0.

* if we add 3 elements index values, that will be equal to 0, then
↓
if this happens, store such index values triplet and return all triplets

* No duplicates allowed

Suppose : $(-1, 0, 1, 2, 2, -1, 4)$

$\text{ans1} = (-1, 0, 1, 2, 2, -1, 4)$ $(-1, 0, 1, 2, 2, -1, 4) = \text{ans2}$

Both answers results in zero, but both triplets are same, so don't allow triplets

simply, for that to happen--

- ① Sort the elements
- ② Skip duplicates

* And, if we have a sorted array, and we are looking for a certain condition on index values



↳ we can use two pointers

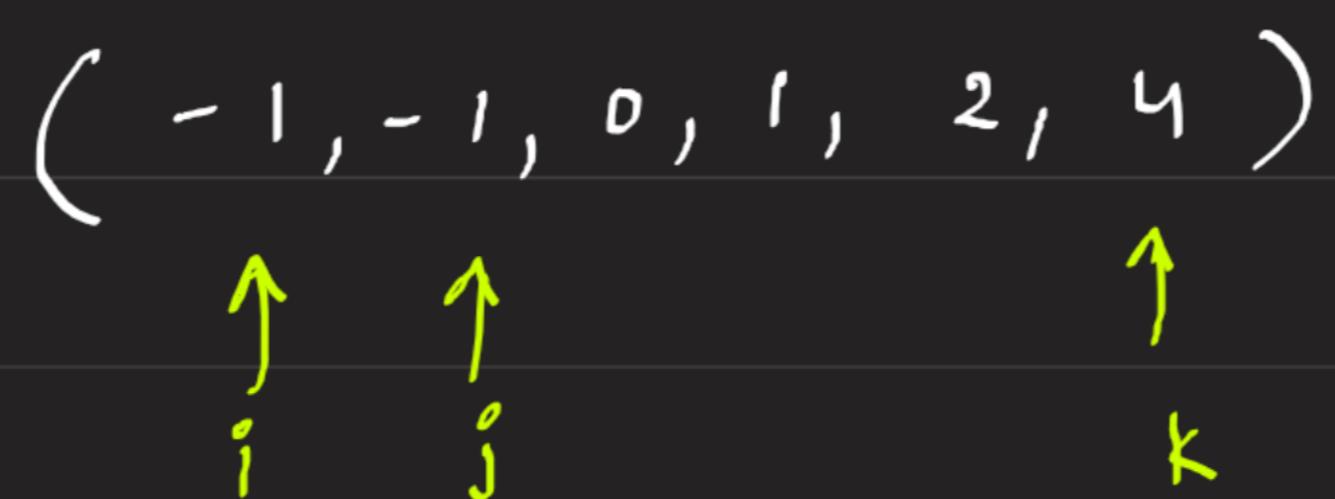
$$\text{nums}[i] + \text{nums}[j] + \text{nums}[k] == 0$$

$\text{nums}[] = (-1, 0, 1, 2, -1, 4)$

↓ sort

$(-1, -1, 0, 1, 2, 4)$

placing pointers :



* We start with $i \rightarrow$ traverse from 0 to $n-2$

* for every $i \rightarrow$ we have (j, k)
↳ j starts from $i+1$
 k starts from $n-1$

Now, apply 2 sum ideology, b/w (j, k) pointers

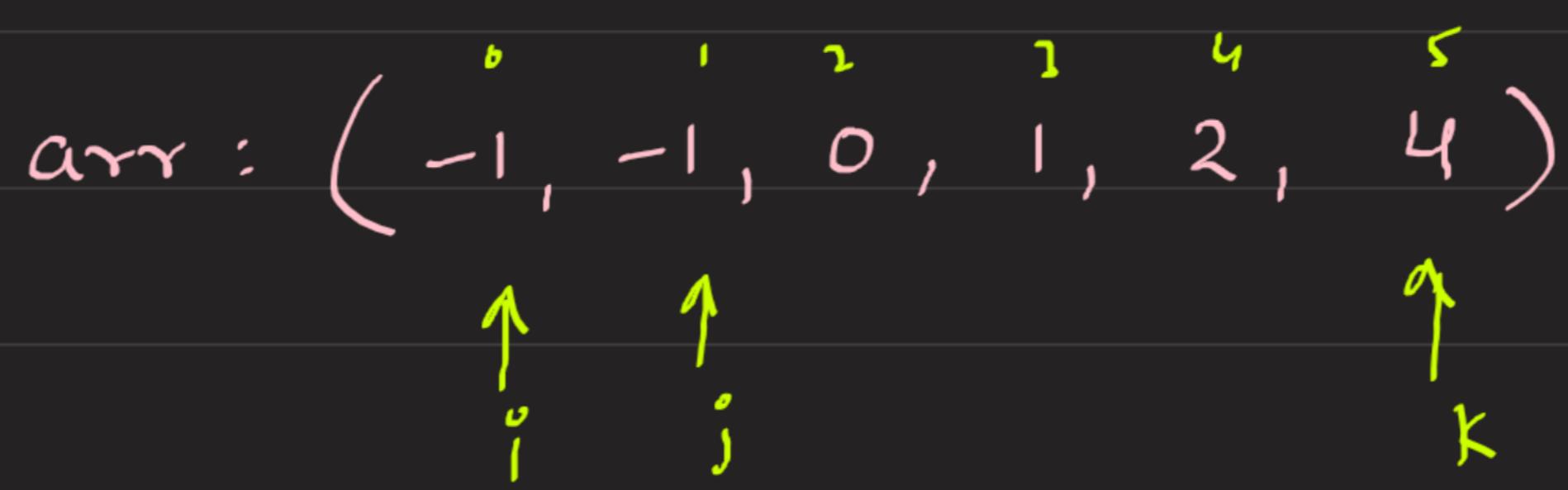
* for every (j, k) look up for $\text{nums}[i] + \text{nums}[j] + \text{nums}[k] == 0$?

if ($\text{sum} < 0$) $\rightarrow j++$ {sum need to increase, so increment j , as array sorted value will increase}

if ($\text{sum} > 0$) $\rightarrow k--$ {sum need to decrease}

if ($\text{sum} == 0$) \rightarrow Add triplet & move j, k

* when (j, k) crosses, move i



i	j	k	Sum	Operation
0	1	5	$2 > 0$	$k--$

i	j	k	Sum	Operation
0	1	4	$0 == 0$	Add triplet

// Skip all $k=2's$ & $j=-1's$ to ignore duplicates

i	j	k	Sum	Operation
0	2	3	$0 == 0$	Add triplet

Because, with $i=0$,

you can't form
any other triplets
with same values
 (j, k)

if ($j & k$ Crosses)

↓
increment $i \rightarrow$ & repeat same

* Above steps are for triplets which includes $i=0$

* Do same steps for $i=1, 2, \dots, n-2$

* Then our ans will consists of every possible triplet

T.C: $O(n \log n)$ \rightarrow sorting

$O(n^2)$ to traverse $(i, j, k) \approx O(n^2)$