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.
```


```python
class Solution:
    def threeSum(self, nums: List[int]) -> List[List[int]]:
        nums.sort()
        res = []

        def twoSum(nums, i, res):
            l, h = i+1, len(nums) -1
            while l < h:
                summ = nums[i] + nums[l] + nums[h]

                if summ < 0:
                    l += 1
                elif summ > 0:
                    h -= 1
                else:
                    res.append([nums[i], nums[l], nums[h]])
                    l += 1
                    h -= 1

                    while l < h and nums[l] == nums[l-1]:
                        l += 1


        for i in range(len(nums)):
            if nums[i] > 0:
                break
            if i == 0 or nums[i-1] != nums[i]:
                twoSum(nums, i, res)
        return res

```