

# LEETCODE SOLUTIONS - YOGESH MUNEEES

**15. 3Sum** Hint

**Medium** 27.6K 2.5K

Companies

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.

## Solution:

```
class Solution:
    def threeSum(self, nums: List[int]) -> List[List[int]]:
        fin = []
        length = len(nums)
        nums.sort()
        for i in range(length-2):
            if i>0 and nums[i]==nums[i-1]:
                continue
            j = i+1
            k = length-1
            while j<k:
                sum3 = nums[i]+nums[j]+nums[k]
                if sum3 == 0:
                    fin.append([nums[i],nums[j],nums[k]])
                    j += 1
                    while j<k and nums[j]==nums[j-1]:
                        j += 1
                elif sum3 > 0:
                    k -= 1
                else:
                    j += 1
        return fin
```