

# 015. 3Sum

## 015 3Sum

- Two Pointers

### Description

Given an array  $S$  of  $n$  integers, are there elements  $a, b, c$  in  $S$  such that  $a + b + c = 0$ ? Find all unique triplets in the array which gives the sum of zero.

**Note:** The solution set must not contain duplicate triplets.

For example, given array  $S = [-1, 0, 1, 2, -1, -4]$ ,

A solution set is:

```
[
  [-1, 0, 1],
  [-1, -1, 2]
]
```

### 1. Thought line

### 2. Two Pointers with optimization

```
1 class Solution {
2 public:
3     vector<vector<int>> threeSum(vector<int>& nums) {
4         vector<vector<int>> result;
5         if (nums.size() < 3) return result;
6
7         int N = nums.size();
8         sort(nums.begin(), nums.end());
9
10        for (int i=0; i<=N-3; ++i){
11            if (i>0 && nums[i]==nums[i-1]) continue;
12            if (nums[i]+nums[N-1]+nums[N-2]<0) continue;
13            if (nums[i]+nums[i+1]+nums[i+2]>0) break;
14
15            int front = i+1, tail = N-1;
16            while (front<tail){
17                if (nums[i] + nums[front] + nums[tail] == 0) {
18                    result.push_back({nums[i], nums[front], nums[tail]});
19                    while (front+1 < tail && nums[front] == nums[front+1])
20                        ++front;
21                    while (tail-1 > front && nums[tail] == nums[tail-1])
22                        --tail;
23                    ++front, --tail;
24                } else if (nums[i] + nums[front] + nums[tail] < 0) ++front;
25                else --tail;
26            }
27        }
28        return result;
29    }
30 };
```