

# 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

```
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums) {
        vector<vector<int>> result;
        if (nums.size() < 3) return result;

        int N = nums.size();
        sort(nums.begin(), nums.end());

        for (int i = 0; i <= N - 3; ++i) {
            if (i > 0 && nums[i] == nums[i - 1]) continue;
            if (nums[i] + nums[N - 1] + nums[N - 2] < 0) continue;
            if (nums[i] + nums[i + 1] + nums[i + 2] > 0) break;

            int front = i + 1, tail = N - 1;
            while (front < tail) {
                if (nums[i] + nums[front] + nums[tail] == 0) {
                    result.push_back({nums[i], nums[front], nums[tail]});
                    while (front + 1 < tail && nums[front] == nums[front + 1])
                        ++front;
                    while (tail - 1 > front && nums[tail] == nums[tail - 1])
                        --tail;
                    ++front, --tail;
                } else if (nums[i] + nums[front] + nums[tail] < 0) ++front;
                else --tail;
            }
        }
        return result;
    }
};
```

