

# 016. 3Sum Closest

## 016 3Sum Closest

- Two Pointers

### Description

Given an array  $S$  of  $n$  integers, find three integers in  $S$  such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

For example, given array  $S = \{-1\ 2\ 1\ -4\}$ , and target = 1.

The sum that is closest to the target is 2.  $(-1 + 2 + 1 = 2)$ .

### Two Pointers

```
1 class Solution {
2 public:
3     int threeSumClosest(vector<int>& nums, int target) {
4
5         if (nums.size() < 3) return 0;
6         sort(nums.begin(), nums.end());
7         int N = nums.size(), sum = 0, dif = INT_MAX;
8         for (int i = 0; i <= N-3; ++i){
9             if (i > 0 && nums[i-1] == nums[i]) continue;
10            int front = i+1, tail = N-1;
11            while(front < tail){
12                int sum_temp = nums[i] + nums[front] + nums[tail];
13                if (sum_temp == target) return target;
14                else{
15                    if (abs(sum_temp - target) < dif){
16                        dif = abs(sum_temp - target);
17                        sum = nums[i] + nums[front] + nums[tail];
18                    }
19                    if(sum_temp < target) {
20                        while(front+1 < tail && nums[front+1] == nums[front]) ++front;
21                        ++front;
22                    }
23                    else{
24                        while(tail-1 > front && nums[tail-1] == nums[tail]) --tail;
25                        --tail;
26                    }
27                }
28            }
29        }
30        return sum;
31    }
32 };
```