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).
```

1. Thought line

2. Two Pointers

```
class Solution {
public:
    int threeSumClosest(vector<int>& nums, int target) {

        if (nums.size() < 3) return 0;
        sort(nums.begin(), nums.end());
        int N = nums.size(), sum = 0, dif = INT_MAX;
        for (int i = 0; i <= N-3; ++i){
            if (i > 0 && nums[i-1] == nums[i]) continue;
            int front = i+1, tail = N-1;
            while(front < tail){
                int sum_temp = nums[i] + nums[front] + nums[tail];
                if (sum_temp == target) return target;
                else{
                    if (abs(sum_temp - target)<dif){
                        dif = abs(sum_temp - target);
                        sum = nums[i] + nums[front] + nums[tail];
                    }
                    if(sum_temp<target) {
                        while(front+1 < tail && nums[front+1] == nums[front]) ++front;
                        ++front;
                    }
                    else{
                        while(tail-1 >front && nums[tail-1] == nums[tail]) --tail;
                        --tail;
                    }
                }
            }
        }
        return sum;
    }
};
```