#LeetCode16

https://leetcode.com/problems/3sum-closest/description/

Yifeng Zeng

#题目描述

3Sum Closest

#思路报告

LC167. Two Sum II - Input array is sorted的思路，左右两个指针，左右之和大于target，那么右边往左移使得sum变小，左右之和小于target，那么左边往右移使得sum变大。本题可以借鉴LC167的思路，3sum无非就是固定一个element，另外两个element像LC167那样移动去找target或者找closest target。那么我们先对数组排序，然后从左往右固定nums[i]，那么其实就是在i后面找最接近target - nums[i]的two sum。也是大于target - nums[i]时，右边往左移使sum变小，小于target - nums[i]时，左边往右移使sum变大。取绝对值来保存different最小时的nums[i]， nums[left], nums[right]的sum。

代码如下：

```
public int threeSumClosest(int[] nums, int target) {
    if (nums == null || nums.length == 0) {
        throw new IllegalArgumentException("Invalid input");
    }

    Arrays.sort(nums);
    int diff = Integer.MAX_VALUE;
    int res = Integer.MAX_VALUE;
    for (int i = 0; i < nums.length - 2; i++) {
        int t = target - nums[i];
        int left = i + 1;
        int right = nums.length - 1;
        while (left < right) {
            int sum = nums[left] + nums[right] - t;
            if (Math.abs(sum) < diff) {
                diff = Math.abs(sum);
                res = sum + target;
            }
```

```
                if (sum > 0) {
                    right--;
                } else {
                    left++;
                }
            }
        }
        return res;
    }
```

#套路总结

- 数组未排序没有好办法解的情况下可以考虑先排序。
- 多个数组合的情况可以考虑固定其中一个。