

16. 3Sum Closest

Description

Hints

Submissions

Discuss

Solution

Pick One

Given an array `nums` of n integers and an integer `target`, find three integers in `nums` such that the sum is closest to `target`. Return the sum of the three integers. You may assume that each input would have exactly one solution.

Example:

Given array `nums = [-1, 2, 1, -4]`, and `target = 1`.

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

```
public class L16 {
    /*
     * 这个和L15 3Sum思路比较相似，首先对数组进行排序，然后设定两个指针left和right，从第一个元素开始处理。
     * 理想情况下，如果找到三个数的和等于target，它们的距离为0，直接输出target。但是我们不能保证数组中一定
     * 存在三个数的和等于target，我们首先维护一个变量des，让它记录三个数的和与target的差值，每次比较des与
     * 当前循环中的current des，如果current des小于全局的des，我们就用result记录当前三个数的和
     */
    public int threeSumClosest(int[] nums, int target) {
        if(nums == null || nums.length < 3)
            return -1;
        int des = Integer.MAX_VALUE;
        int result = 0;
        Arrays.sort(nums);

        for(int i = 0; i < nums.length; i++) {
            int l = i + 1;
            int r = nums.length - 1;
            while (l < r) {
                int sum = nums[i] + nums[l] + nums[r];
                if(sum == target) {
                    return target;
                } else if (sum < target) {
                    if(target - sum < des){
                        des = target - sum;
                        result = sum;
                    }
                    l++;
                } else {
                    if(sum - target < des) {
                        des = sum - target;
                        result = sum;
                    }
                    r--;
                }
            }
        }
        return result;
    }
}
```