

最接近的三数之和

给定一个包括 n 个整数的数组 `nums` 和一个目标值 `target`。找出 `nums` 中的三个整数，使得它们的和与 `target` 最接近。返回这三个数的和。假定每组输入只存在唯一答案。

示例：

输入：nums = [-1,2,1,-4], target = 1

输出：2

解释：与 target 最接近的和是 2 (-1 + 2 + 1 = 2)。

```
class Solution {
public:
    int threeSumClosest(vector<int>& nums, int target) {
        if(nums.size()<3)
            return -1;
        int res=-1e9;//初始极小值方便后续比较 需要考虑范围 INT_MIN 超过范围
        sort(nums.begin(),nums.end());
        for(int i=0;i<nums.size();i++)
        {
            if(i>0&&nums[i]==nums[i-1])
                continue;
            int left=i+1;
            int right=nums.size()-1;
            while(left<right)
            {
                int sum=nums[i]+nums[left]+nums[right];
                if(sum==target)
                    return target;
                else if(sum<target)
                    left++;
                else if(sum>target)
                    right--;
                else
                {
                    while(left!=i+1&&nums[left]==nums[left-1])
                        left++;
                    while(right!=nums.size()-1&&nums[right]==nums[right+1])
                        right--;
                }
                res=abs(target-res)>abs(target-sum)?sum:res;
            }
        }
    }
}
```

```
        return res;  
    }  
};
```