

18. 四数之和

Tuesday, July 26, 2022 10:33 AM

<https://leetcode.cn/problems/4sum/>

中等

给你一个由 n 个整数组成的数组 $nums$ ，和一个目标值 $target$ 。请你找出并返回满足下述全部条件且不重复的四元组 $[nums[a], nums[b], nums[c], nums[d]]$

示例 1:

输入: $nums = [1,0,-1,0,-2,2]$, $target = 0$

输出: $[[-2,-1,1,2],[-2,0,0,2],[-1,0,0,1]]$

示例 2:

输入: $nums = [2,2,2,2,2]$, $target = 8$

输出: $[[2,2,2,2]]$

思路：在三数之和外再套for循环。

```
class Solution {
    public List<List<Integer>> fourSum(int[] nums, int target) {
        List<List<Integer>> res=new ArrayList<>();
        Arrays.sort(nums);

        for(int i=0;i<nums.length;i++){
            if(nums[i]>0 && nums[i]>target){ // 测试用例有
[1000000000,1000000000,1000000000,1000000000] -294967296
                return res;
            }
            if(i>0 && nums[i]==nums[i-1]){ // i重复了
                continue;
            }
            for(int j=i+1;j<nums.length;j++){
                if(j>i+1 && nums[j]==nums[j-1]){ // j重复了
                    continue;
                }
                int left=j+1;
                int right=nums.length-1;
                while(left<right){
                    long sum=nums[i]+nums[j]+nums[left]+nums[right];
                    if(sum>target){
                        right--;
                    }else if(sum<target){
                        left++;
                    }else{
                        res.add(Arrays.asList(nums[i],nums[j],nums[left],nums[right]));
                        while(left<right && nums[right]==nums[right-1]) right--; //去重
                        while(left<right && nums[left]==nums[left+1]) left++; // 去重
                        left++;
                    }
                }
            }
        }
        return res;
    }
}
```

```
        right--;  
    }  
    }  
    }  
    }  
    return res;  
}  
}
```