

统计特殊四元组

给你一个下标从 0 开始的整数数组 `nums`，返回满足下述条件的不同四元组 (a, b, c, d) 的数目：

- $nums[a] + nums[b] + nums[c] == nums[d]$ ，且
- $a < b < c < d$

示例 1：

输入：`nums = [1,2,3,6]`

输出：1

解释：满足要求的唯一一个四元组是 $(0, 1, 2, 3)$ 因为 $1 + 2 + 3 == 6$ 。

示例 2：

输入：`nums = [3,3,6,4,5]`

输出：0

解释：`[3,3,6,4,5]` 中不存在满足要求的四元组。

示例 3：

输入：`nums = [1,1,1,3,5]`

输出：4

解释：满足要求的 4 个四元组如下：

- $(0, 1, 2, 3): 1 + 1 + 1 == 3$
- $(0, 1, 3, 4): 1 + 1 + 3 == 5$
- $(0, 2, 3, 4): 1 + 1 + 3 == 5$
- $(1, 2, 3, 4): 1 + 1 + 3 == 5$

```
class Solution:
    def countQuadruplets(self, nums: List[int]) -> int:
        count=0
        for a in range(len(nums)):
            for b in range(a+1,len(nums)):
                for c in range(b+1,len(nums)):
                    for d in range(c+1,len(nums)):
```

```
        if nums[a]+nums[b]+nums[c]==nums[d]:  
            count+=1  
    return count
```