

# Problem 1995: Count Special Quadruplets

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 63.99%

**Paid Only:** No

**Tags:** Array, Hash Table, Enumeration

## Problem Description

Given a \*\*0-indexed\*\* integer array `nums`, return \_the number of\*\*distinct\*\* quadruplets\_ `(a, b, c, d)` \_such that:\_

\* `nums[a] + nums[b] + nums[c] == nums[d]` , and \* `a < b < c < d`

**Example 1:**

**Input:** nums = [1,2,3,6] **Output:** 1 **Explanation:** The only quadruplet that satisfies the requirement is (0, 1, 2, 3) because  $1 + 2 + 3 == 6$ .

**Example 2:**

**Input:** nums = [3,3,6,4,5] **Output:** 0 **Explanation:** There are no such quadruplets in [3,3,6,4,5].

**Example 3:**

**Input:** nums = [1,1,1,3,5] **Output:** 4 **Explanation:** The 4 quadruplets that satisfy the requirement are: - (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):  $1 + 1 + 3 == 5$

**Constraints:**

\* `4 <= nums.length <= 50` \* `1 <= nums[i] <= 100`

## Code Snippets

### C++:

```
class Solution {  
public:  
    int countQuadruplets(vector<int>& nums) {  
  
    }  
};
```

### Java:

```
class Solution {  
    public int countQuadruplets(int[] nums) {  
  
    }  
}
```

### Python3:

```
class Solution:  
    def countQuadruplets(self, nums: List[int]) -> int:
```