# 5863. Count Special Quadruplets

My Submissions (/contest/weekly-contest-257/problems/count-special-quadruplets/submissions/)

Back to Contest (/contest/weekly-contest-257/)

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`

| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Easy |

**Example 1:**

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

**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`

JavaScript ▾

```
 1▾ const countQuadruplets = (a) => {
 2      let n = a.length, res = 0;
 3      // a.sort((x, y) => x - y);
 4▾     for (let i = 0; i < n; i++) {
 5▾         for (let j = i + 1; j < n; j++) {
 6▾             for (let k = j + 1; k < n; k++) {
 7▾                 for (let m = k + 1; m < n; m++) {
 8▾                     if (a[i] + a[j] + a[k] == a[m]) {
 9                          // pr(i, j, k, m)
10                          // pr(a[i], a[j], a[k], a[m])
11                          res++;
12                      }
13                  }
14              }
15          }
```

```
16        }
17        return res;
18  };
```

☐ **Custom Testcase**      ( Use Example Testcases )

⏵ Run       ☁ Submit

**Submission Result:** Accepted (/submissions/detail/549627553/) ❓        More Details ❯ (/submissions/detail/549627553/)

Share your acceptance!

---