

2824. Count Pairs Whose Sum is Less than Target

Easy  528  50  Add to List  Share

Given a **0-indexed** integer array `nums` of length `n` and an integer `target`, return the number of pairs (i, j) where $0 \leq i < j < n$ and $nums[i] + nums[j] < target$.

Example 1:

Input: `nums = [-1,1,2,3,1]`, `target = 2`
Output: `3`
Explanation: There are 3 pairs of indices that satisfy the conditions in the statement:
- $(0, 1)$ since $0 < 1$ and $nums[0] + nums[1] = 0 < target$
- $(0, 2)$ since $0 < 2$ and $nums[0] + nums[2] = 1 < target$
- $(0, 4)$ since $0 < 4$ and $nums[0] + nums[4] = 0 < target$
Note that $(0, 3)$ is not counted since $nums[0] + nums[3]$ is not strictly less than the target.

Example 2:

Input: `nums = [-6,2,5,-2,-7,-1,3]`, `target = -2`
Output: `10`
Explanation: There are 10 pairs of indices that satisfy the conditions in the statement:
- $(0, 1)$ since $0 < 1$ and $nums[0] + nums[1] = -4 < target$
- $(0, 3)$ since $0 < 3$ and $nums[0] + nums[3] = -8 < target$
- $(0, 4)$ since $0 < 4$ and $nums[0] + nums[4] = -13 < target$
- $(0, 5)$ since $0 < 5$ and $nums[0] + nums[5] = -7 < target$
- $(0, 6)$ since $0 < 6$ and $nums[0] + nums[6] = -3 < target$
- $(1, 4)$ since $1 < 4$ and $nums[1] + nums[4] = -5 < target$
- $(3, 4)$ since $3 < 4$ and $nums[3] + nums[4] = -9 < target$
- $(3, 5)$ since $3 < 5$ and $nums[3] + nums[5] = -3 < target$
- $(4, 5)$ since $4 < 5$ and $nums[4] + nums[5] = -8 < target$
- $(4, 6)$ since $4 < 6$ and $nums[4] + nums[6] = -4 < target$

Constraints:

- $1 \leq \text{nums.length} == n \leq 50$
- $-50 \leq \text{nums}[i], \text{target} \leq 50$

Accepted 127.9K Submissions 146.9K

```
1 class Solution:
2     def countPairs(self, nums: List[int], target: int) -> int:
3         count = 0
4         for i in range(len(nums)):
5             for j in range(i + 1, len(nums)):
6                 if nums[i] + nums[j] < target:
7                     count += 1
8         return count
```

Your previous code was restored from your local storage. [Reset to default](#)

NEW