

2824. Count Pairs Whose Sum is Less than Target

0-indexed **nums** **n** **target** *the number of*
pairs (i, j) where $0 \leq i < j < n$ and $\text{nums}[i] + \text{nums}[j] < \text{target}$

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 $\text{nums}[0] + \text{nums}[1] = 0 < \text{target}$
- (0, 2) since $0 < 2$ and $\text{nums}[0] + \text{nums}[2] = 1 < \text{target}$
- (0, 4) since $0 < 4$ and $\text{nums}[0] + \text{nums}[4] = 0 < \text{target}$

Note that (0, 3) is not counted since $\text{nums}[0] + \text{nums}[3]$ 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 $\text{nums}[0] + \text{nums}[1] = -4 < \text{target}$
- (0, 3) since $0 < 3$ and $\text{nums}[0] + \text{nums}[3] = -8 < \text{target}$
- (0, 4) since $0 < 4$ and $\text{nums}[0] + \text{nums}[4] = -13 < \text{target}$
- (0, 5) since $0 < 5$ and $\text{nums}[0] + \text{nums}[5] = -7 < \text{target}$
- (0, 6) since $0 < 6$ and $\text{nums}[0] + \text{nums}[6] = -3 < \text{target}$
- (1, 4) since $1 < 4$ and $\text{nums}[1] + \text{nums}[4] = -5 < \text{target}$
- (3, 4) since $3 < 4$ and $\text{nums}[3] + \text{nums}[4] = -9 < \text{target}$
- (3, 5) since $3 < 5$ and $\text{nums}[3] + \text{nums}[5] = -3 < \text{target}$
- (4, 5) since $4 < 5$ and $\text{nums}[4] + \text{nums}[5] = -8 < \text{target}$
- (4, 6) since $4 < 6$ and $\text{nums}[4] + \text{nums}[6] = -4 < \text{target}$

Constraints:

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

2824. Count Pairs Whose Sum is Less than Target

```
/*
    Sorting+Binary search
    Time complexity:  $O(n \log n)$ 
    Space complexity:  $O(1)$ 
*/
class Solution {
public:
    int countPairs(std::vector<int>& nums, int target){
        std::sort(nums.begin(),nums.end());
        int n=nums.size();

        int ans=0;
        for(int i=0;i<n-1;++i){
            // Find the in the range  $[i+1,n-1]$   $x$ ,
            // such that  $nums[i]+x<target \Rightarrow nums[i]+x\leq target-1 \Rightarrow x\leq target-nums[i]-1$ 
            int j=std::upper_bound(
                nums.begin()+i+1,
                nums.end(),
                target-nums[i]-1)-nums.begin()-1; // -1 to get the index of the last value of  $x\leq target-nums[i]-1$ 

            ans+= (j-i);
        }
        return ans;
    }
};
```