**Problem Name: Two Sum- Greater than target**

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

**Output Format:**

**Constraints:**

**Examples:**

**Brute force Solution:**

# Explanation:

**Code:**

**Time Complexity**: O(NlogN)

**Space Complexity:** O(n)

**Optimized Solution:**

# Explanation:

Algorithm: Double Pointer

Sort the array first

Traverse from both sides to the middle using double pointers

- Using monotonicity, if nums[l] + nums[r] > target, then nums[l + k] + nums[r] > target (k >= 0)

- each time the answer is calculated result += r - l means that nums[l] can be paired with nums[l + 1 : r + 1] these r - l elements are paired and greater than target

code ideas

Sort (ascending or descending order, this step is to make the array monotonic)

Set left and right pointers l = 0, r = n - 1

If the sum of the elements pointed to by l and the elements pointed to by r is not greater than target, that is, nums[l] + nums[r] < target, then l is shifted to the right to increase nums[l]; otherwise, result += r - l, and r shift left

Repeat step 3 until l >= r

**Code:**

**Time Complexity**: O(nlogn)

**Space Complexity:** O(1)