Problem 2563: Count the Number of Fair Pairs

Problem Information

Difficulty: Medium
Acceptance Rate: 52.75%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Sorting

Problem Description

Given a **0-indexed** integer array `nums` of size `n` and two integers `lower` and `upper`, return _the number of fair pairs_.

A pair `(i, j)` is **fair** if:

* `0 <= i < j < n`, and * `lower <= nums[i] + nums[j] <= upper`

Example 1:

Input: nums = [0,1,7,4,4,5], lower = 3, upper = 6 **Output:** 6 **Explanation:** There are 6 fair pairs: (0,3), (0,4), (0,5), (1,3), (1,4), and (1,5).

Example 2:

Input: nums = [1,7,9,2,5], lower = 11, upper = 11 **Output:** 1 **Explanation:** There is a single fair pair: (2,3).

Constraints:

* `1 <= nums.length <= 105` * `nums.length == n` * `-109 <= nums[i] <= 109` * `-109 <= lower <= upper <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long countFairPairs(vector<int>& nums, int lower, int upper) {

}
};
```

Java:

```
class Solution {
public long countFairPairs(int[] nums, int lower, int upper) {

}
}
```

Python3:

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