Problem 532: K-diff Pairs in an Array

Problem Information

Difficulty: Medium
Acceptance Rate: 45.31%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Binary Search, Sorting

Problem Description

Given an array of integers `nums` and an integer `k`, return _the number of**unique** k-diff pairs in the array_.

A **k-diff** pair is an integer pair `(nums[i], nums[j])`, where the following are true:

* `0 <= i, j < nums.length` * `i != j` * `|nums[i] - nums[j]| == k`

Notice that `|val|` denotes the absolute value of `val`.

Example 1:

Input: nums = [3,1,4,1,5], k = 2 **Output:** 2 **Explanation:** There are two 2-diff pairs in the array, (1, 3) and (3, 5). Although we have two 1s in the input, we should only return the number of **unique** pairs.

Example 2:

Input: nums = [1,2,3,4,5], k = 1 **Output:** 4 **Explanation:** There are four 1-diff pairs in the array, (1, 2), (2, 3), (3, 4) and (4, 5).

Example 3:

Input: nums = [1,3,1,5,4], k = 0 **Output:** 1 **Explanation:** There is one 0-diff pair in the array, (1, 1).

Constraints:

* `1 <= nums.length <= 104` * `-107 <= nums[i] <= 107` * `0 <= k <= 107`

Code Snippets

C++:

```
class Solution {
public:
int findPairs(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int findPairs(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def findPairs(self, nums: List[int], k: int) -> int:
```