Problem 2465: Number of Distinct Averages

Problem Information

Difficulty: Easy
Acceptance Rate: 58.77%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Sorting

Problem Description

You are given a **0-indexed** integer array `nums` of **even** length.

As long as `nums` is **not** empty, you must repetitively:

* Find the minimum number in `nums` and remove it. * Find the maximum number in `nums` and remove it. * Calculate the average of the two removed numbers.

The **average** of two numbers `a` and `b` is `(a + b) / 2`.

* For example, the average of `2` and `3` is `(2 + 3) / 2 = 2.5`.

Return _the number of**distinct** averages calculated using the above process_.

Note that when there is a tie for a minimum or maximum number, any can be removed.

Example 1:

Input: nums = [4,1,4,0,3,5] **Output:** 2 **Explanation:** 1. Remove 0 and 5, and the average is (0 + 5) / 2 = 2.5. Now, nums = [4,1,4,3]. 2. Remove 1 and 4. The average is (1 + 4) / 2 = 2.5, and nums = [4,3]. 3. Remove 3 and 4, and the average is (3 + 4) / 2 = 3.5. Since there are 2 distinct numbers among 2.5, 2.5, and 3.5, we return 2.

Example 2:

Input: nums = [1,100] **Output:** 1 **Explanation:** There is only one average to be calculated after removing 1 and 100, so we return 1.

Constraints:

* `2 <= nums.length <= 100` * `nums.length` is even. * `0 <= nums[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int distinctAverages(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int distinctAverages(int[] nums) {

}
}
```

Python3:

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