Problem 1674: Minimum Moves to Make Array Complementary

Problem Information

Difficulty: Medium
Acceptance Rate: 42.90%
Paid Only: No
Tags: Array, Hash Table, Prefix Sum

Problem Description

You are given an integer array `nums` of **even** length `n` and an integer `limit`. In one move, you can replace any integer from `nums` with another integer between `1` and `limit`, inclusive.

The array `nums` is **complementary** if for all indices `i` (**0-indexed**), `nums[i] + nums[n - 1 - i]` equals the same number. For example, the array `[1,2,3,4]` is complementary because for all indices `i`, `nums[i] + nums[n - 1 - i] = 5`.

Return the _**minimum** number of moves required to make _`nums` _**complementary**_.

Example 1:

Input: nums = [1,2,4,3], limit = 4 **Output:** 1 **Explanation:** In 1 move, you can change nums to [1,2,_2_ ,3] (underlined elements are changed). nums[0] + nums[3] = 1 + 3 = 4. nums[1] + nums[2] = 2 + 2 = 4. nums[2] + nums[1] = 2 + 2 = 4. nums[3] + nums[0] = 3 + 1 = 4. Therefore, nums[i] + nums[n-1-i] = 4 for every i, so nums is complementary.

Example 2:

Input: nums = [1,2,2,1], limit = 2 **Output:** 2 **Explanation:** In 2 moves, you can change nums to [_2_ ,2,2,_2_]. You cannot change any number to 3 since 3 > limit.

Example 3:

Input: nums = [1,2,1,2], limit = 2 **Output:** 0 **Explanation:** nums is already complementary.

Constraints:

* `n == nums.length` * `2 <= n <= 105` * `1 <= nums[i] <= limit <= 105` * `n` is even.

Code Snippets

C++:

```
class Solution {
public:
int minMoves(vector<int>& nums, int limit) {

}
};
```

Java:

```
class Solution {
public int minMoves(int[] nums, int limit) {

}
}
```

Python3:

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