4.20 COUNTING GOOD PAIRS IN AN INTEGER ARRAY

Question:

Given an array of integers nums, return the number of good pairs. A pair (i, j) is called good if nums[i] == nums[j] and i < j.

AIM

To count the number of good pairs in an array where elements match and the first index is less than the second.

ALGORITHM

1. Initialize a dictionary count to store the frequency of each number.
2. Traverse the array and for each number:
    - Add the current count of that number to the result (since each previous occurrence forms a good pair with the current one).
    - Increment the count of that number.
3. Return the total number of good pairs.

PROGRAM

```
def num_good_pairs(nums):
    count = {}
    result = 0
    for num in nums:
        result += count.get(num, 0)
        count[num] = count.get(num, 0) + 1
    return result

nums = list(map(int, input("Enter numbers separated by space: ").split()))
print("Number of good pairs:", num_good_pairs(nums))
```

Input:

Enter numbers separated by space: 1 2 3 1 3 2 3

Output:

```
Enter numbers separated by space: 1 2 3 1 3 2 3
Number of good pairs: 5
>>>
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: O(n), where n is the length of the array
- Space Complexity: O(n)