**EX-1.3**

**Title** :

You are given a 0-indexed integer array nums. The distinct count of a subarray of nums is defined as: Let nums[i..j] be a subarray of nums consisting of all the indices from i to j such that 0 <= i <= j < nums.length. Then the number of distinct values in nums[i..j] is called the distinct count of nums[i..j]. Return the sum of the squares of distinct counts of all subarrays of nums. A subarray is a contiguous non-empty sequence of elements within an array.

**Aim**:

To design and implement a Python program that calculates the sum of the squares of the count of distinct elements for every subarray of a given integer array.

**Procedure**

1. Take input for the integer array `nums` from the user.
2. Generate all possible subarrays of `nums`.
3. For each subarray, count the number of distinct elements.
4. Square that distinct count and add it to a running sum.
5. Output the resulting sum after processing all subarrays.

## Algorithm

1. **Start**
2. Read the input integer array `nums`.
3. Initialize `result` to 0.
4. For `i` in range from 0 to length of `nums` - 1:
    - Create an empty set `distinct_set`.
    - For `j` in range from `i` to length of `nums` - 1:
        - Add `nums[j]` to `distinct_set`.
        - Calculate the size of `distinct_set`.
        - Add the square of this size to `result`.
5. Print or return `result`.
6. **End**

**Input:**

Enter elements of the array (space separated): 1 2 1

**Output:**

Sum of squares of distinct counts of all subarrays: 15

**Program :**

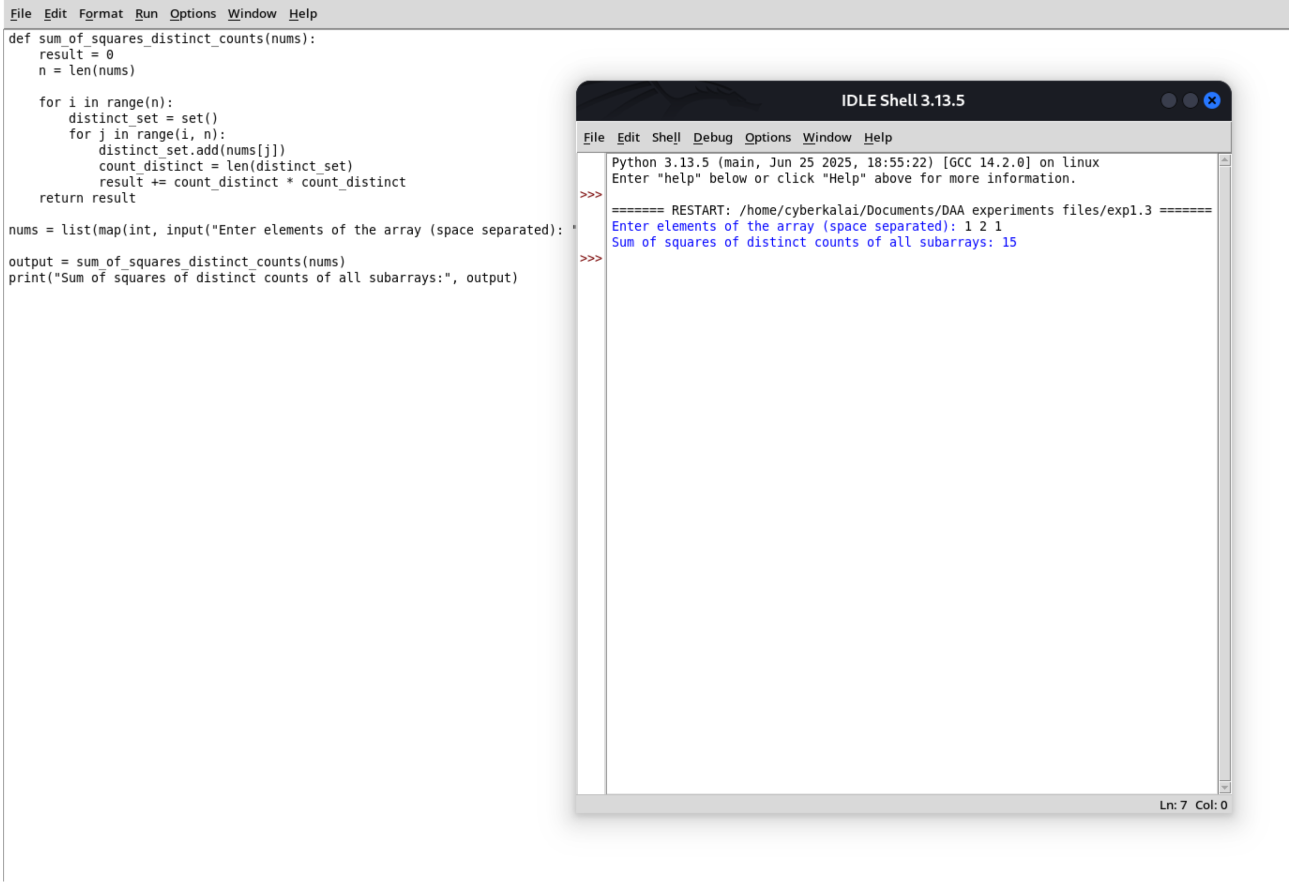
```
def sum_of_squares_distinct_counts(nums):
    result = 0
    n = len(nums)

    for i in range(n):
        distinct_set = set()
        for j in range(i, n):
            distinct_set.add(nums[j])
            count_distinct = len(distinct_set)
            result += count_distinct * count_distinct
    return result
nums = list(map(int, input("Enter elements of the array (space separated): ").split()))
output = sum_of_squares_distinct_counts(nums)
print("Sum of squares of distinct counts of all subarrays:", output)
```

**Performance Analysis:**

Time complexity : $O(n^2)$

Space complexity: $O(n)$

## program output:



File Edit Format Run Options Window Help
IDLE Shell 3.13.5
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0

```
def sum_of_squares_distinct_counts(nums):
    result = 0
    n = len(nums)

    for i in range(n):
        distinct_set = set()
        for j in range(i, n):
            distinct_set.add(nums[j])
            count_distinct = len(distinct_set)
            result += count_distinct * count_distinct
    return result

nums = list(map(int, input("Enter elements of the array (space separated): '

output = sum_of_squares_distinct_counts(nums)
print("Sum of squares of distinct counts of all subarrays:", output)
```

```
Python 3.13.5 (main, Jun 25 2025, 18:55:22) [GCC 14.2.0] on linux
Enter "help" below or click "Help" above for more information.
>>>
======= RESTART: /home/cyberkalai/Documents/DAA experiments files/exp1.3 =======
Enter elements of the array (space separated): 1 2 1
Sum of squares of distinct counts of all subarrays: 15
>>>
```

## Result :

Thus the given program to find the Sum of Squares of Distinct Counts of All Subarrays is executed and got output successfully.