**1.3 .**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:**

Sum of Squares of Distinct Counts of All Subarrays

**ALGORITHM:**

· Initialize answer = 0.

· For each starting index i (0 → n-1):

* Create a frequency map (hashmap).
* For each ending index j (i → n-1):
  + Insert nums[j] into the map (track counts).
  + The number of distinct elements = size of map.
  + Add (distinct\_count)^2 to answer.

· Return answer.

**PROGRAM:**

def sum\_of\_squares\_distinct(nums):

n = len(nums)

total = 0

for i in range(n):

seen = set()

for j in range(i, n):

seen.add(nums[j])

distinct\_count = len(seen)

total += distinct\_count \*\* 2

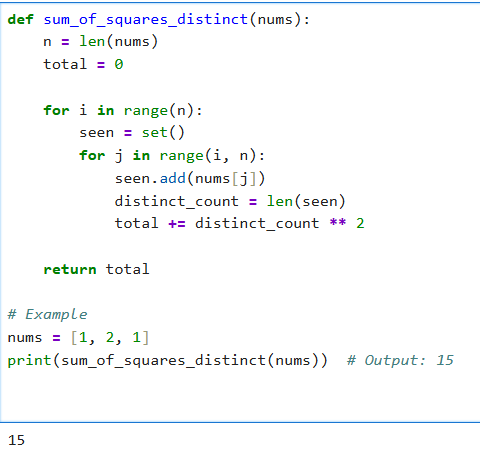
return total

# Example

nums = [1, 2, 1]

print(sum\_of\_squares\_distinct(nums))

**INPUT AND OUTPUT:**



**RESULT:**

Thus the program for Sum of Squares of Distinct Counts of All Subarrays executed successfully and output is verified.