

## **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  $0 \leq i \leq j < \text{nums.length}$ . 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.

## **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:
5. Create an empty set `distinct_set`.
6. For `j` in range from `i` to length of `nums` - 1:
7. Add `nums[j]` to `distinct_set`.
8. Calculate the size of `distinct_set`.
9. Add the square of this size to result.
10. Print or return result.
11. **End**

## **Input:**

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

## **Output:**

Sum of squares of distinct counts of all subarrays: 15

## **Performance Analysis:**

Time complexity :  $O(n^2)$

Space complexity:  $O(n)$

## **program output:**

```
main.py
1 - def sumOfSquaresDistinct(nums):
2 -     n = len(nums)
3 -     total = 0
4 -
5 -     for i in range(n):
6 -         seen = set()
7 -         for j in range(i, n):
8 -             seen.add(nums[j])
9 -             distinct = len(seen)
10 -            total += distinct * distinct
11 -    return total
12 |
13 print(sumOfSquaresDistinct([1,2,1]))
14 print(sumOfSquaresDistinct([1,1]))
15 print(sumOfSquaresDistinct([1,2,3]))
```

Output

```
15
3
20
==== Code Execution Successful ===
```

**Result :**

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