

1. 126. 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.

Example 1:

Input: `nums = [1,2,1]`

Output: 15

Code:

```
def sum_of_squares_of_distinct_counts(nums):
    from collections import defaultdict

    n = len(nums)
    total_sum = 0

    for left in range(n):
        count_map = defaultdict(int)
        distinct_count = 0
        for right in range(left, n):
            if count_map[nums[right]] == 0:
                distinct_count += 1
            count_map[nums[right]] += 1
            total_sum += distinct_count ** 2

    return total_sum

nums = [1, 2, 1]
print(sum_of_squares_of_distinct_counts(nums))
```

output:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/problem.py
15
PS C:\Users\karth>
```

Time complexity: $f(n) = O(n^2)$