


992. Subarrays with K Different Integers

Solved 

Hard

 Topics

 Companies

Given an integer array `nums` and an integer `k`, return *the number of **good subarrays** of `nums`*.

A **good array** is an array where the number of different integers in that array is exactly `k`.

- For example, `[1,2,3,1,2]` has `3` different integers: `1`, `2`, and `3`.

A **subarray** is a **contiguous** part of an array.

Example 1:

Input: `nums = [1,2,1,2,3]`, `k = 2`

Output: `7`

Explanation: Subarrays formed with exactly 2 different integers: `[1,2]`, `[2,1]`, `[1,2]`, `[2,3]`, `[1,2,1]`, `[2,1,2]`, `[1,2,1,2]`

Sliding Window

```
class Solution:
```

```
    def subarraysWithKDistinct(self, nums: List[int], k: int) ->
int:
```

```
        count = defaultdict(int)
```

```
        ans = 0
```

```
        l_far = 0
```

```
        l_near = 0
```

```
        # r = 0
```

```
        for r in range(len(nums)):
```

```
            count[nums[r]] += 1
```

```
            while(len(count) > k):
```

```
                count[nums[l_near]] -= 1
```

```
                if count[nums[l_near]] == 0:
```

```
                    count.pop(nums[l_near])
```

```
                l_near += 1
```

```
                l_far = l_near
```

```
            while(count[nums[l_near]] > 1):
```

```
                count[nums[l_near]] -= 1
```

```
l_near += 1
```

```
if(len(count) == k):
```

```
    ans = ans + l_near - l_far + 1
```

```
return ans
```