[LeetCode](https://leetcode.com/problems/length-of-longest-subarray-with-at-most-k-frequency/description/?envType=daily-question&envId=2024-03-28)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/length-of-longest-subarray-with-at-most-k-frequency/description/?envType=daily-question&envId=2024-03-28>

<https://www.linkedin.com/in/aditya-adi-konda/>

Day 38 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Ventured further into my coding journey today, tackling the engaging LeetCode Problem "Successfully solved LeetCode Problem ���"

“2958. Length of Longest Subarray With at Most K Frequency.”

   
✨ Task: You are given an integer array nums and an integer k.

The frequency of an element x is the number of times it occurs in an array.

An array is called good if the frequency of each element in this array is less than or equal to k.

Return the length of the longest good subarray of nums.

A subarray is a contiguous non-empty sequence of elements within an array.

Examples:

Example 1:

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

Output: 6

Explanation: The longest possible good subarray is [1,2,3,1,2,3] since the values 1, 2, and 3 occur at most twice in this subarray. Note that the subarrays [2,3,1,2,3,1] and [3,1,2,3,1,2] are also good.

It can be shown that there are no good subarrays with length more than 6.

Example 2:

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

Output: 2

Explanation: The longest possible good subarray is [1,2] since the values 1 and 2 occur at most once in this subarray. Note that the subarray [2,1] is also good.

It can be shown that there are no good subarrays with length more than 2.

Example 3:

Input: nums = [5,5,5,5,5,5,5], k = 4

Output: 4

Explanation: The longest possible good subarray is [5,5,5,5] since the value 5 occurs 4 times in this subarray.

It can be shown that there are no good subarrays with length more than 4.

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public int maxSubarrayLength(int[] nums, int k) {

    int ans = 0;

    Map<Integer, Integer> count = new HashMap<>();

    for (int l = 0, r = 0; r < nums.length; ++r) {

      count.merge(nums[r], 1, Integer::sum);

      while (count.get(nums[r]) == k + 1)

        count.merge(nums[l++], -1, Integer::sum);

      ans = Math.max(ans, r - l + 1);

    }

    return ans;

  }

}

