**Equivalent Sub-Arrays**

Show Topic Tags   

Given an array of n integers. Count the total number of sub-array having total distinct elements same as that of total distinct elements of the original array.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Each test case contains  
the number of elements in the array a[] as n and next line contains space separated n elements in  
the array.

**Output:**  
For each test case output an integer which is the required answer.

**Constraints:**  
1<=T<=100  
1<=n<=100  
1<=a[i]<=100

**Example:  
Input:**  
3  
5  
2 1 3 2 3  
5  
2 4 5 2 1  
5  
2 4 4 2 4  
**Output:**  
5  
2  
9

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/equivalent-sub-arrays/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication241;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.HashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication241 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String[] input = br.readLine().trim().split(" ");

int[] arr = new int[n];

for(int i =0; i<n; i++) {

arr[i] = Integer.parseInt(input[i]);

}

//int[] arr = { 2, 1, 3, 2, 3};

//int[] arr = {2, 4, 4, 2, 4};

HashSet<Integer> todos = new HashSet<Integer>();

for(int i =0; i<arr.length; i++) {

todos.add(arr[i]);

}

System.out.println(todos.size());

int cont =0;

HashSet<Integer> hs= new HashSet<Integer>();

for(int i =0; i<arr.length; i++) {

hs = new HashSet<Integer>();

hs.add(arr[i]);

for(int j=i+1; j<arr.length; j++) {

hs.add(arr[j]);

if(hs.size() == todos.size()) {

cont++;

}

}

}

if(hs.size() == todos.size()) {

cont++;

}

System.out.println(cont);

}

}

}

--------------solucion optima------------------

<http://www.geeksforgeeks.org/count-subarrays-total-distinct-elements-original-array/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int countDistictSubarray(int[] arr, int n)

{

// Count distinct elements in whole array

Dictionary<int, int> vis = new Dictionary<int, int>();

for (int i = 0; i < n; ++i)

{

if (!vis.ContainsKey(arr[i]))

{

vis.Add(arr[i], 1);

}

}

int k = vis.Count;

// Reset the container by removing all elements

vis.Clear();

// Use sliding window concept to find

// count of subarrays having k distinct

// elements.

int ans = 0, right = 0, window = 0;

for (int left = 0; left < n; ++left)

{

while (right < n && window < k)

{

if (vis.ContainsKey(arr[right]))

{

vis[arr[right]]++;

}

else

{

vis[arr[right]] = 1;

}

if (vis.ContainsKey(arr[right]) && vis[arr[right]] == 1)

++window;

++right;

}

// If window size equals to array distinct

// element size, then update answer

if (window == k)

ans += (n - right + 1);

// Decrease the frequency of previous element

// for next sliding window

if (vis.ContainsKey(arr[left]))

{

vis[arr[left]]--;

}

// If frequency is zero then decrease the

// window size

if (vis.ContainsKey(arr[left]) && vis[arr[left]] == 0)

--window;

}

return ans;

}

static void Main(string[] args)

{

int[] arr = { 2, 1, 3, 2, 3};

//int[] arr = {2, 4, 4, 2, 4};

//int[] arr = { 73 };

Console.WriteLine(countDistictSubarray(arr, arr.Length));

Console.ReadLine();

}

}

}