https://course.acciojob.com/idle?question=e4db4e3a-7e09-4168-8a84-df5187f34ca3

- HARD
- Max Score: 50 Points

Subarrays With Distinct Integers

Given an array `a` of positive integers. A subarray of `a` is considered good if the number of different integers in that subarray is exactly `b`.

Count the number of good subarrays of `a`.

Input Format

First line contains the size of array `n`.

Second line contains n-spaced integers representing array `a`.

Third line contains an integer representing `b`.

Output Format

Print an integer denoting the number of good subarrays.

Example 1

Input

```
5
1 2 1 2 3
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].

Example 2

Input

```
5
1 2 1 3 4
3
```

Output

```
3
```

Explanation

Subarrays formed with exactly 3 different integers: [1, 2, 1, 3], [2, 1, 3], [1, 3, 4].

Constraints

1 <= n <= 40000

1 <= a[i] <= n

1 <= b <= n

Topic Tags

- **Hashing**
- **2-Pointers**
- **Arrays**

My code

import java.util.*;

```
class Accio {
            static int atMostK(int[] A, int K) {
        int i = 0, res = 0;
                    //i is left and j will be right pointer
        Map<Integer, Integer> count = new HashMap<>();
        for (int j = 0; j < A.length; ++j)
        {
            if (count.getOrDefault(A[j], 0) == 0) K--;
                                //k-- ie we hahe to assign it ie one new item assign in hm
            count.put(A[j], count.getOrDefault(A[j], 0) + 1);
            while (K < 0) //ie more than k item present
            {
                count.put(A[i], count.get(A[i]) - 1);
                if (count.get(A[i]) == 0) K++;//ie one item left from hm
                i++;
            }
            res += j - i + 1;//per pointer new no of sub aray
        }
        return res;
}
    public static int solve(int[] a, int b) {
        //Your code goes here
                     return atMostK(a, b) - atMostK(a, b - 1);
    }

}
public class Main {
  public static void main (String[] args)
            {
                    Scanner sc = new Scanner(System.in);

                int n = sc.nextInt();
                    int[] a = new int[n];
            for(int i=0;i<n;i++)
                    {
                                a[i] = sc.nextInt();
                    }
                    int b = sc.nextInt();
                    Accio Obj = new Accio();
        System.out.println(Obj.solve(a, b));
        sc.close();
            }
}
```