The Land of Mines - Hackerearth

Tuesday, May 19, 2020 7:49 PM

As our mighty hero Sin is in pursuit of the wicked wizard,what he faces ahead is the land of mines .
In order to cross the land of mines Sin needs to disarm all the mines in his path.Individual mine contained an array of integers and a separate individual integer k written on it and a space above which "key" is written.
In order to diffuse a mine Sin needs to find key.The key to diffuse the bomb resides in the array written on individual mines.
To diffuse a mine Sin needs to find the count of all the sub arrays which contains k or more than k similar elements.
Since Sin is weak at calculations help Sin calculate the key in order to diffuse individual mine.

Contraints
$1<=t<=1000$
$1<=n<=10^5$
$1<=a_i<=10^9$
$1<=k<=10^5$

Input
First line contains integer t denoting number of testcases , Next t testcases contains :
FIrst line contains integer n (size of array) and k
Second line contains array of integers.
Output
for every testcase , output no. of subarrays.
SAMPLE INPUT

2
7 1
2 4 2 4 3 1 2
4 2
4 3 4 3
SAMPLE OUTPUT

28
3

Explanation
For test case 2 : the sub arrays are: {4,3,4},{4,3,4,3},{3,4,3}

The Code:

```
#include <stdio.h>
#define LEN 100000
int main() {
    int caseCount, N, k, left, right, prevLeft, index, leftIndex;
    long long int currCount, leftCombi, rightCombi, result;
```

```
    int arr[LEN], counts[LEN];

    scanf("%d", &caseCount);
    while(caseCount > 0) {
        scanf("%d %d", &N, &k);
        if(k == 1) {
            result = N * (N+1) / 2;
            printf("%lld\n", result);
            for(left=0;left<N;left++) {
                scanf("%d", &arr[left]); // Just exhausting the inputs. What a
waste of time.
            }
            caseCount--;
            continue;
        }
        for(left=0; left<LEN; left++) {
            counts[left] = 0;
        }
        prevLeft = -1; currCount = right = left = result = 0;
        for(right=0; right<N; right++) {
            scanf("%d", &arr[right]);

            index = arr[right];
            counts[index] += 1;

            if(counts[index] >= k) {

                for(left; left<right; left++) {
                    leftIndex = arr[left];
                    if(arr[left] != arr[right]) {
                        counts[leftIndex] -= 1;
                    } else if(counts[index] > k){
                        counts[index] -= 1;
                    } else {
                        break;
                    }
                }

                leftCombi = (left-prevLeft);
                rightCombi = (N-right);
                result += leftCombi * rightCombi;
                prevLeft = left;
            }
        }
        printf("%lld\n", result);
        caseCount--;
    }
}
```

The Stats:

Score
30.0

Time (sec)
1.31631

Memory (KiB)
976

Language
C