Codeforces Round #492

Problema C: *Sonya and Robots*

Prof. Edson Alves – UnB/FGA

Problema

Since Sonya is interested in robotics too, she decided to construct robots that will read and recognize numbers.

Sonya has drawn n numbers in a row, a_i is located in the i-th position. She also has put a robot at each end of the row (to the left of the first number and to the right of the last number). Sonya will give a number to each robot (they can be either same or different) and run them. When a robot is running, it is moving toward to another robot, reading numbers in the row. When a robot is reading a number that is equal to the number that was given to that robot, it will turn off and stay in the same position.

Sonya does not want robots to break, so she will give such numbers that robots will stop before they meet. That is, the girl wants them to stop at different positions so that the first robot is to the left of the second one.

Problema

For example, if the numbers $[1, 5, 4, 1, 3]$ are written, and Sonya gives the number 1 to the first robot and the number 4 to the second one, the first robot will stop in the 1-st position while the second one in the 3-rd position. In that case, robots will not meet each other. As a result, robots will not be broken. But if Sonya gives the number 4 to the first robot and the number 5 to the second one, they will meet since the first robot will stop in the 3-rd position while the second one is in the 2-nd position.

Sonya understands that it does not make sense to give a number that is not written in the row because a robot will not find this number and will meet the other robot.

Problema

Sonya is now interested in finding the number of different pairs that she can give to robots so that they will not meet. In other words, she wants to know the number of pairs (p, q), where she will give p to the first robot and q to the second one. Pairs (p_i, q_i) and (p_j, q_j) are different if $p_i \neq p_j$ or $q_i \neq q_j$.

Unfortunately, Sonya is busy fixing robots that broke after a failed launch. That is why she is asking you to find the number of pairs that she can give to robots so that they will not meet.

Entrada e saída

Input

The first line contains a single integer n $(1 \leq n \leq 10^5)$ – the number of numbers in a row.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^5)$ – the numbers in a row.

Output

Print one number – the number of possible pairs that Sonya can give to robots so that they will not meet.

Exemplo de entradas e saídas

Sample Input

```
5
1 5 4 1 3

7
1 2 1 1 1 3 2
```

Sample Output

```
9

7
```

Solução com complexidade $O(N \log N)$

- Para computar o número de pares (a, b) distintos, é preciso observar que a pode assumir apenas valores distintos
- Para cada a, b pode assumir todos os valores distintos que estão à direita da primeira ocorrência de a no vetor
- Um conjunto pode ser usado manter o registro dos valores de a já processados
- O registro do valores à direita de a pode ser mantido através do uso de um histograma
- Com estas duas estruturas auxiliares, para cada elemento a_i do vetor, na ordem dada na entrada, são três passos a serem realizados
- O primeiro é a atualização do histograma, removendo o valor a_i se for a última ocorrência no vetor
- O segundo é verificar se a_i já foi processado ou não: em caso positivo, deve-se seguir para o próximo valor
- Por fim, se a_i não foi processado, devem ser acumulados na resposta todos os valores distintos ainda presentes no histograma

Solução AC com complexidade $O(N \log N)$

```cpp
#include <bits/stdc++.h>

using namespace std;

long long solve(const vector<int>& as)
{
    map<int, int> R;                // Histograma à direita
    set<int> processed;
    long long ans = 0;

    for (auto a : as)
        ++R[a];

    for (auto a : as)
    {
        --R[a];

        if (R[a] == 0)
            R.erase(a);
```

Solução AC com complexidade $O(N \log N)$

```
21          if (processed.count(a))
22              continue;
23
24          ans += R.size();            // Soma os elementos distintos à direita
25          processed.insert(a);
26      }
27
28      return ans;
29  }
```

Solução AC com complexidade $O(N \log N)$

```
31 int main()
32 {
33     ios::sync_with_stdio(false);
34
35     int N;
36     cin >> N;
37
38     vector<int> as(N);
39
40     for (int i = 0; i < N; ++i)
41         cin >> as[i];
42
43     auto ans = solve(as);
44
45     cout << ans << '\n';
46
47     return 0;
48 }
```