Codeforces Beta Round #98

Problema B: *Permutation*

Prof. Edson Alves – UnB/FGA

Problema

“Hey, it’s homework time” – thought Polycarpus and of course he started with his favourite subject, IT. Polycarpus managed to solve all tasks but for the last one in 20 minutes. However, as he failed to solve the last task after some considerable time, the boy asked you to help him.

The sequence of n integers is called a permutation if it contains all integers from 1 to n exactly once.

You are given an arbitrary sequence $a_1, a_2, \ldots, a_n$ containing n integers. Each integer is not less than 1 and not greater than 5000. Determine what minimum number of elements Polycarpus needs to change to get a permutation (he should not delete or add numbers). In a single change he can modify any single sequence element (i. e. replace it with another integer).

Entrada e saída

Input

The first line of the input data contains an integer n $(1 \leq n \leq 5000)$ which represents how many numbers are in the sequence. The second line contains a sequence of integers a_i $(1 \leq a_i \leq 5000, 1 \leq i \leq n)$.

Output

Print the only number – the minimum number of changes needed to get the permutation.

Exemplo de entradas e saídas

Sample Input

```
3
3 1 2

2
2 2

5
5 3 3 3 1
```

Sample Output

```
0


1


2
```

Solução com complexidade $O(N)$

- É possível resolver este problema com complexidade $O(N^2)$: basta, para cada valor $i = 1, 2, \ldots, N$, percorrer todo o vetor em busca deste valor
- Se o valor não for localizado, basta incrementar a resposta
- Contudo, há uma solução com complexidade $O(N)$
- Basta criar um vetor auxiliar v com $N + 1$ elementos, todos iguais a zero
- Para cada elemento a do vetor da entrada, faça $v_a = 1$
- O número de elementos a serem alterados é igual ao total N, subtraído do número de encontrados (a soma de todos os valores armazenados em v)

Solução AC com complexidade $O(N)$

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

using namespace std;

int solve(int N, vector<int>& as)
{
    vector<int> found(N + 1, 0);

    for (const auto& a : as)
        if (1 <= a and a <= N)
            found[a] = 1;

    int total = accumulate(found.begin(), found.end(), 0);

    return N - total;
}
```

Solução AC com complexidade $O(N)$

```cpp
int main()
{
    ios::sync_with_stdio(false);

    int N;
    cin >> N;

    vector<int> as(N);

    for (int i = 0; i < N; ++i)
        cin >> as[i];

    cout << solve(N, as) << '\n';

    return 0;
}
```