Conjunto Permutado

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Descripción**

Dado dos conjuntos A y B de 1<=N<=1000 elementos enteros cada uno, determina si A es una permutación de B, es decir, si A tiene los mismos elementos que B (aunque estén distinto orden)

**Entrada**

En la primera línea el entero N que denota el número de elementos en cada conjunto. En la segunda línea, los N enteros del conjunto A separados por espacios. En la tercera línea, los N enteros del conjunto B separados por espacios.

**Salida**

Imprime SI (con mayúsculas) si A es una permutación de B, o NO, en caso contrario.

**Ejemplo**

|  |  |
| --- | --- |
| Entrada  4  3 2 7 1  2 1 3 7 | Salida  SI |

*Fuente: lacj*

Problema subido por: [lacj20](https://omegaup.com/profile/lacj20/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Conjunto-Permutado#problems/new-run) | | | | | | | | |
| 7dad4cc2 | Respuesta correcta | 100.00% | 0 | cpp | 3.05 | 0.01 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Conjunto-Permutado#problems>

#include <stdio.h>

#include <vector>

#include <map>

#include <iostream>

using namespace std;

void esPermutacion(int a[], int b[], int n) {

std::map<int,int> map\_a;

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

int key = a[i];

if(map\_a.count(key)) {

map\_a[key]++;

}else {

map\_a[key] = 1;

}

}

std::map<int,int> map\_b;

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

int key = b[i];

if(map\_b.count(key)) {

map\_b[key]++;

} else {

map\_b[key] = 1;

}

}

/\*

for (std::map<char,int>::iterator it=mymap.begin(); it!=mymap.end(); ++it)

std::cout << it->first << " => " << it->second << '\n';

\*/

for(std::map<int,int>::iterator it = map\_a.begin(); it != map\_a.end(); it++) {

//std::cout << it->first << " => " << it->second << '\n';

if(it->second != map\_b[it->first]) {

printf("NO");

return ;

}

}

printf("SI");

}

int main() {

int N;

scanf("%d",&N);

int a[N];

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

scanf("%d",&a[i]);

}

int b[N];

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

scanf("%d", &b[i]);

}

esPermutacion(a, b, N);

return 0;

}