

Truco de magia

Contribución de Gaston Fontenla Nuñez

Descripción del problema

Un mago quiere realizar un truco de magia. Para eso utilizará N vasos colocados boca abajo sobre una mesa. Cada uno de los vasos tiene escrito un número entre 0 y $N - 1$ inclusive. Debajo de cada uno hay una cierta cantidad de bolitas de telgopor, que varía de vaso en vaso.

Utilizando uno de los K “hechizos” que hay en su libro de magia, el mago puede intercambiar sin que nadie lo note las bolitas que hay debajo de **un determinado par**. El hechizo i permite al mago intercambiar las bolitas del vaso numerado a_i con las del numerado b_i .

Una vez aprendido por el mago, cada hechizo puede utilizarse durante el truco tantas veces como sea necesario, pero el mago no puede intercambiar las cantidades de bolitas en un par de vasos para el cual no haya aprendido un hechizo: sus habilidades no se lo permiten.

El truco de magia estelar consiste en realizar una serie de intercambios “usando la mente”, y asombrando así a todo el público. Al final del truco, el mago revela la cantidad de bolitas que han quedado debajo de cada vaso.

Debes escribir una función que dadas las cantidades de bolitas iniciales, el listado de hechizos en el libro de magia, y las cantidades finales de bolitas deseada, le indique al mago si puede llevar a cabo el truco o no.

Aprender muchos hechizos diferentes es desgastante para el mago. De ser posible, le encantaría realizar el truco sin tener que recordar más que N hechizos diferentes.

Como desafío adicional, la función debe calcular un conjunto de **a lo sumo N hechizos** (de entre los K dados) que le permita completar el truco con éxito.

Te garantizamos que, en aquellos casos de prueba en que sea posible realizar el truco, también será posible realizarlo utilizando como máximo N hechizos diferentes.

Detalles de implementación

Debes implementar la función `mago(cantInicial, cantFinal, a, b, hechizos)` que retorne verdadero (`true`) si es posible llevar a cabo el truco, y falso (`false`) en caso contrario.

Los parámetros de la función son:

- Un arreglo de N enteros `cantInicial`. El elemento i indica la cantidad de bolitas que hay inicialmente debajo del vaso con número i . $0 \leq i \leq N - 1$.
- Un arreglo de N enteros `cantFinal`. El elemento i indica la cantidad de bolitas que deberán quedar al finalizar el truco debajo del vaso con número i . $0 \leq i \leq N - 1$.
- Arreglos `a` y `b`, ambos de K elementos, que indican los pares de vasos para los que existe un hechizo. Para cada i entre 0 y $K - 1$ inclusive, el hechizo i permite intercambiar las cantidades de bolitas debajo de los vasos `a[i]` y `b[i]`.
- Un arreglo `hechizos`, donde almacenar los índices de los hechizos elegidos para que aprenda el mago. Deben ser a lo sumo N , sin repetir, y cada uno deberá estar entre 0 y $K - 1$ inclusive.

Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con el entero N
- Una línea con los N enteros `cantInicial`
- Una línea con los N enteros `cantFinal`
- Una línea con el entero K
- K líneas. La i -ésima de estas líneas (contando desde 0) contiene dos enteros $a[i]$ y $b[i]$

Escribe a la salida estándar el valor retornado por la función `mago` (en formato SI o NO), así como el valor del arreglo hechizos.

Cotas

- $2 \leq N \leq 100.000$
- $1 \leq K \leq 200.000$
- $1 \leq \text{cantInicial}[i] \leq 1.000.000.000$
- $1 \leq \text{cantFinal}[i] \leq 1.000.000.000$
- $0 \leq a[i] < b[i] \leq N - 1$
- Para todo par $0 \leq i < j \leq N - 1$, se tendrá que $(a[i], b[i]) \neq (a[j], b[j])$. En otras palabras, los K pares dados serán diferentes.

Puntuación

Se obtiene el 60% del puntaje por la respuesta "SI/NO" correcta, y el 40% restante por indicar correctamente un conjunto de a lo sumo N hechizos para aprender.

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
4
5 40 9 1
9 5 40 1
5
0 1
1 2
0 2
0 3
1 3
```

Para un programa correcto, la salida podría ser:

```
SI
0 2 1 4
```

Si en cambio recibe la siguiente entrada:

```
3
1 2 3
3 2 1
1
0 1
```

Para un programa correcto, la salida debe ser:

```
NO
```

Subtareas

- $N \leq 8$ (15 puntos)
- $N \leq 500$ (20 puntos)
- Todos los valores de `cantInicial` serán iguales a 2, excepto por un único i tal que `cantInicial[i] = 1`. De la misma forma, en `cantFinal` habrá algún j tal que `cantFinal[j] = 1`, y todos los demás valores serán 2 (30 puntos)
- Sin más restricciones (35 puntos)