

Jeroglíficos

Un equipo de investigadores está estudiando las similitudes entre secuencias de jeroglíficos. Ellos representan cada jeroglífico con un entero no negativo. Para realizar su estudio, ellos usan los siguientes conceptos acerca de secuencias.

Para una secuencia fija A , una secuencia S es llamada una **subsecuencia** de A si y solo si S puede ser obtenida removiendo algunos de los elementos de A (posiblemente ninguno).

La siguiente tabla muestra algunos ejemplos de subsecuencias de la secuencia $A = [3, 2, 1, 2]$.

Subsecuencia	Cómo puede obtenerse a partir de A
$[3, 2, 1, 2]$	Ningún elemento es removido.
$[2, 1, 2]$	$[3, 2, 1, 2]$
$[3, 2, 2]$	$[3, 2, \cancel{1}, 2]$
$[3, 2]$	$[3, \cancel{2}, \cancel{1}, 2]$ o $[3, 2, \cancel{1}, \cancel{2}]$
$[3]$	$[3, \cancel{2}, \cancel{1}, \cancel{2}]$
$[\]$	$[\cancel{3}, \cancel{2}, \cancel{1}, \cancel{2}]$

Por otra parte, $[3, 3]$ o $[1, 3]$ no son subsecuencias de A .

Considere dos secuencias de jeroglíficos, A y B . Una secuencia S es llamada **subsecuencia común** de A y B si y solo si S es una subsecuencia de ambas A y B . Adicionalmente, decimos que una secuencia U es una **subsecuencia común universal** de A y B si y solo si se cumplen las siguientes dos condiciones:

- U es una subsecuencia común de A y B .
- Toda subsecuencia común de A y B es también una subsecuencia de U .

Se puede demostrar que para cualquier dos subsecuencias A y B , estas tienen a lo sumo una subsecuencia común universal.

Los investigadores han encontrado dos secuencias de jeroglíficos A y B . La secuencia A consiste de N jeroglíficos y la secuencia B consiste de M jeroglíficos. Ayude a los investigadores a computar una subsecuencia común universal de A y B , o determine si tal secuencia no existe.

Detalles de Implementación

Debes implementar la siguiente función.

```
std::vector<int> ucs(std::vector<int> A, std::vector<int> B)
```

- A : arreglo de longitud N que describe la primera secuencia.
- B : arreglo de longitud M que describe la segunda secuencia.
- En caso de existir una subsecuencia común universal de A y B , la función debe retornar un arreglo que contenga dicha secuencia. Caso contrario, la función debe retornar $[-1]$ (un arreglo de longitud 1, cuyo único elemento es -1).
- Esta función es llamada exactamente una vez por cada caso de prueba.

Restricciones

- $1 \leq N \leq 100\,000$
- $1 \leq M \leq 100\,000$
- $0 \leq A[i] \leq 200\,000$ para cada i tal que $0 \leq i < N$
- $0 \leq B[j] \leq 200\,000$ para cada j tal que $0 \leq j < M$

Subtareas

Subtarea	Puntaje	Restricciones Adicionales
1	3	$N = M$; Cada una de A y B consisten de N enteros distintos entre 0 y $N - 1$ (inclusive)
2	15	Para cada entero k , $a + b \leq 3$, donde a es la cantidad de veces que aparece k en A y b es la cantidad de veces que aparece k en B .
3	10	$A[i] \leq 1$ para cada i tal que $0 \leq i < N$; $B[j] \leq 1$ para cada j tal que $0 \leq j < M$
4	16	Existe una subsecuencia común universal de A y B .
5	14	$N \leq 3000$; $M \leq 3000$
6	42	Sin restricciones adicionales.

Ejemplos

Ejemplo 1

Considere la siguiente llamada.

```
ucs([0, 0, 1, 0, 1, 2], [2, 0, 1, 0, 2])
```

En este caso, las subsecuencias comunes de A y B son las siguientes: $[], [0], [1], [2], [0, 0], [0, 1], [0, 2], [1, 0], [1, 2], [0, 0, 2], [0, 1, 0], [0, 1, 2], [1, 0, 2]$ y $[0, 1, 0, 2]$.

Ya que $[0, 1, 0, 2]$ es una subsecuencia común de A y B , y todas las subsecuencias comunes de A y B son subsecuencias de $[0, 1, 0, 2]$, la función debe retornar $[0, 1, 0, 2]$.

Ejemplo 2

Considere la siguiente llamada.

```
ucs([0, 0, 2], [1, 1])
```

En este caso, la única subsecuencia común de A y B es la secuencia vacía $[]$. Por lo tanto, la función debe retornar un arreglo vacío $[]$.

Ejemplo 3

Considere la siguiente llamada.

```
ucs([0, 1, 0], [1, 0, 1])
```

En este caso, las subsecuencias comunes de A y B son $[], [0], [1], [0, 1]$ y $[1, 0]$. Se puede demostrar que una subsecuencia común universal no existe. Por lo tanto, la función debe retornar $[-1]$.

Calificador local

Formato de entrada:

```
N M
A[0] A[1] ... A[N-1]
B[0] B[1] ... B[M-1]
```

Formato de salida:

```
T
R[0] R[1] ... R[T-1]
```

En este caso, R es el arreglo retornado por `ucs` y T es su longitud.