

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 A , una secuencia S es una **subsecuencia** de A si y solo si S puede ser obtenida eliminando algunos elementos (es posible no eliminar ninguno) de A .

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

Subsecuencia	Cómo puede ser obtenida desde A
$[3, 2, 1, 2]$	No se elimina ningún elemento.
$[2, 1, 2]$	$[3, 2, 1, 2]$
$[3, 2, 2]$	$[3, 2, \cancel{1}, 2]$
$[3, 2]$	$[3, \cancel{2}, \cancel{1}, 2]$ or $[3, 2, \cancel{1}, \cancel{2}]$
$[3]$	$[3, \cancel{2}, \cancel{1}, \cancel{2}]$
$[]$	$[3, \cancel{2}, \cancel{1}, \cancel{2}]$

Por otro lado, $[3, 3]$ o $[1, 3]$ no son subsecuencias de A .

Considera dos secuencias de jeroglíficos, A y B . Una secuencia S es una **subsecuencia común** de A y B si y solo si S es una subsecuencia tanto de A como de B . Además de eso, 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 es una subsecuencia de U .

Se puede probar que para cualquier par de secuencias A y B hay a los más 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. Ayuda a los investigadores a computar una subsecuencia común universal de las secuencias A y B , o determine que no existe tal secuencia.

Detalles de Implementación

Debes implementar la siguiente función.

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

- A : un arreglo de tamaño N que describe la primera secuencia.
- B : un arreglo de tamaño M que describe la segunda secuencia.
- Si existe una subsecuencia común universal de A y B , la función debe retornar un arreglo que contenga esta secuencia. De lo contrario, la función debe retornar $[-1]$ (un arreglo de tamaño 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

Subtareas	Puntos	Restricciones Adicionales
1	3	$N = M$; cada una de las secuencias A y B consisten de N enteros distintos entre 0 y $N - 1$ (inclusive)
2	15	Para cualquier entero k , (el número de elementos de A iguales a k) sumado a (el número de elementos de B iguales a k) es a lo más 3.
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

Considera la siguiente llamada.

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

Aquí, 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 $[0, 1, 0, 2]$.

Como $[0, 1, 0, 2]$ 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

Considera la siguiente llamada.

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

Aquí, la única subsecuencia común de A y B es la secuencia vacía $[]$. Por eso la función debe retornar un arreglo vacío $[]$.

Ejemplo 3

Considera la siguiente llamada.

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

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

Evaluador de prueba

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]
```

Aquí, R es el arreglo retornado por ucs y T es su tamaño.