

# Hieroglyphs

Uma equipa de investigadores está a estudar as semelhanças entre sequências de hieróglifos. Eles representam cada hieróglifo com um inteiro não-negativo. Para realizarem o seu estudo, eles usam os seguintes conceitos sobre sequências.

Para uma dada sequência  $A$ , uma sequência  $S$  é chamada de **subsequência** de  $A$  se e só se  $S$  puder ser obtida através de remoção de alguns elementos (possivelmente nenhum) de  $A$ .

A tabela seguinte mostra alguns exemplos de subsequências de uma sequência  $A = [3, 2, 1, 2]$ .

Subsequência	Como pode ser obtida a partir de $A$
$[3, 2, 1, 2]$	Nenhum elemento é removido.
$[2, 1, 2]$	$[3, 2, 1, 2]$
$[3, 2, 2]$	$[3, 2, 1, 2]$
$[3, 2]$	$[3, 2, 1, 2]$ or $[3, 2, 1, 2]$
$[3]$	$[3, 2, 1, 2]$
$[]$	$[3, 2, 1, 2]$

Por outro lado,  $[3, 3]$  ou  $[1, 3]$  não são subsequências de  $A$ .

Considera duas sequências de hieróglifos,  $A$  e  $B$ . A sequência  $S$  é chamada de **subsequência comum** de  $A$  e  $B$  se e só se  $S$  for uma subsequência tanto de  $A$  como de  $B$ . Mais do que isso, dizemos que uma sequência  $U$  é uma **subsequência comum universal** de  $A$  e  $B$  se e só se as seguintes duas condições forem cumpridas:

- $U$  é uma subsequência comum de  $A$  e  $B$ .
- Todas as subsequências comuns de  $A$  e  $B$  são também subsequências de  $U$ .

Pode ser mostrado que quaisquer duas sequências  $A$  e  $B$  têm no máximo uma subsequência comum universal.

Os investigadores descobriram duas sequências de hieróglifos  $A$  e  $B$ . A sequência  $A$  consiste em  $N$  hieróglifos e a sequência  $B$  consiste em  $M$  hieróglifos. Ajuda os investigadores a calcular a subsequência comum universal de  $A$  e  $B$ , ou determina que essa subsequência não existe.

## Detalhes de Implementação

Deves implementar a seguinte função.

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

- $A$ : array de tamanho  $N$  descrevendo a primeira sequência.
- $B$ : array de tamanho  $M$  descrevendo a segunda sequência.
- Se existir uma subsequência comum universal de  $A$  e  $B$ , a função deve devolver um array contendo essa sequência. Caso contrário, a função deve devolver  $[-1]$  (um array de tamanho 1, cujo único elemento é  $-1$ ).
- Esta função é chamada exatamente uma vez para cada caso de teste.

## Restrições

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

## Subtarefas

Subtarefa	Pontos	Restrições Adicionais
1	3	$N = M$ ; cada uma das sequências $A$ e $B$ consiste $N$ em inteiros <b>distintos</b> entre 0 e $N - 1$ (inclusive)
2	15	Para qualquer inteiro $k$ , a soma entre (o número de elementos de $A$ iguais a $k$ ) e (o número de elementos de $B$ iguais a $k$ ) é no máximo 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 uma subsequência comum universal de $A$ e $B$ .
5	14	$N \leq 3000$ ; $M \leq 3000$
6	42	Nenhuma restrição adicional.

## Exemplos

### Exemplo 1

Considera a seguinte chamada.

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

Aqui, as subsequências de  $A$  e  $B$  são as seguintes:  $[], [0], [1], [2], [0,0], [0,1], [0,2], [1,0], [1,2], [0,0,2], [0,1,0], [0,1,2], [1,0,2]$  e  $[0,1,0,2]$

Como  $[0,1,0,2]$  é uma subsequência comum de  $A$  e  $B$ , e todas subsequências comuns de  $A$  e  $B$  são subsequências de  $[0,1,0,2]$ , a função deve devolver  $[0,1,0,2]$ .

## Exemplo 2

Considera a seguinte chamada.

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

Aqui, a única subsequência comum de  $A$  e  $B$  é a sequência vazia  $[]$ . Daqui decorre que a função deve devolver um array vazio  $[]$ .

## Exemplo 3

Considera a seguinte chamada.

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

Aqui, as subsequências comuns de  $A$  e  $B$  são  $[], [0], [1], [0,1]$  e  $[1,0]$ . Pode ser mostrado que uma subsequência comum universal não existe. Portanto, a função deve devolver  $[-1]$ .

## Avaliador Exemplo

Formato de input:

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

Formato de output:

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

Aqui,  $R$  é o array devolvido por `ucs` e  $T$  o seu tamanho.