

Hieroglif

Sebuah tim peneliti sedang mempelajari kesamaan di antara sekuens-sekuens hieroglif. Mereka merepresentasikan setiap hieroglif dengan sebuah bilangan bulat non-negatif. Untuk melakukan penelitiannya, mereka menggunakan konsep mengenai sekuens sebagai berikut.

Untuk sebuah sekuens tetap A , sekuens S dikatakan sebagai **subsekuens** dari A jika dan hanya jika S dapat diperoleh dengan menghapus beberapa elemen (bisa saja nol) dari A .

Tabel berikut menunjukkan beberapa contoh subsekuens dari sekuens $A = [3, 2, 1, 2]$.

Subsekuens	Cara memperolehnya dari A
$[3, 2, 1, 2]$	Tidak ada elemen yang dihapus.
$[2, 1, 2]$	$[\cancel{3}, 2, 1, 2]$
$[3, 2, 2]$	$[3, 2, \cancel{1}, 2]$
$[3, 2]$	$[3, \cancel{2}, \cancel{1}, 2]$ atau $[3, 2, \cancel{1}, \cancel{2}]$
$[3]$	$[3, \cancel{2}, \cancel{1}, \cancel{2}]$
$[\]$	$[\cancel{3}, \cancel{2}, \cancel{1}, \cancel{2}]$

Di sisi lain, $[3, 3]$ atau $[1, 3]$ bukanlah subsekuens dari A .

Perhatikan dua sekuens hieroglif A dan B . Sekuens S disebut sebagai **subsekuens persekutuan** dari A dan B jika dan hanya jika S adalah subsekuens dari A dan B . Kemudian, sekuens U disebut sebagai **subsekuens persekutuan semesta** dari A dan B jika dan hanya jika dua syarat berikut terpenuhi:

- U adalah sebuah subsekuens persekutuan dari A dan B .
- Setiap subsekuens persekutuan dari A dan B merupakan subsekuens dari U .

Bisa dibuktikan bahwa dua sekuens A dan B hanya mempunyai paling banyak satu subsekuens persekutuan semesta.

Tim peneliti tersebut menemukan dua sekuens hieroglif A dan B . Sekuens A terdiri dari N hieroglif dan sekuens B terdiri dari M hieroglif. Bantulah tim peneliti tersebut mencari sebuah subsekuens persekutuan semesta dari sekuens A dan B , atau lapor jika sekuens tersebut tidak mungkin ada.

Detail Implementasi

Anda harus mengimplementasikan prosedur berikut.

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

- A : array sepanjang N yang mendeskripsikan sekuens pertama.
- B : array sepanjang M yang mendeskripsikan sekuens kedua.
- Jika terdapat sebuah subsekuens persekutuan semesta dari A dan B , prosedur ini mengembalikan sebuah array berisi sekuens tersebut. Jika tidak, prosedur ini harus mengembalikan $[-1]$ (sebuah array sepanjang 1, dengan -1 sebagai elemen satu-satunya).
- Prosedur ini dipanggil tepat sekali untuk setiap kasus uji.

Batasan

- $1 \leq N \leq 100\,000$
- $1 \leq M \leq 100\,000$
- $0 \leq A[i] \leq 200\,000$ untuk setiap i sehingga $0 \leq i < N$
- $0 \leq B[j] \leq 200\,000$ untuk setiap j sehingga $0 \leq j < M$

Subsoal

Subsoal	Nilai	Batasan Tambahan
1	3	$N = M$; A dan B masing-masing terdiri dari N bilangan bulat berbeda dari 0 hingga $N - 1$ (inklusif)
2	15	Untuk bilangan bulat sembarang k , (banyaknya elemen di A yang sama dengan k) ditambah dengan (banyaknya elemen di B yang sama dengan k) tidak lebih dari 3.
3	10	$A[i] \leq 1$ untuk setiap i sehingga $0 \leq i < N$; $B[j] \leq 1$ untuk setiap j sehingga $0 \leq j < M$
4	16	Terdapat sebuah subsekuens persekutuan semesta dari A dan B .
5	14	$N \leq 3000$; $M \leq 3000$
6	42	Tidak ada batasan tambahan.

Contoh

Contoh 1

Perhatikan pemanggilan berikut.

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

Di sini, subsekuens persekutuan dari A dan B adalah sebagai berikut: $[], [0], [1], [2], [0, 0], [0, 1], [0, 2], [1, 0], [1, 2], [0, 0, 2], [0, 1, 0], [0, 1, 2], [1, 0, 2]$ dan $[0, 1, 0, 2]$.

Karena $[0, 1, 0, 2]$ adalah sebuah subsekuens persekutuan dari A dan B , dan semua subsekuens persekutuan dari A dan B adalah subsekuens dari $[0, 1, 0, 2]$, prosedur harus mengembalikan $[0, 1, 0, 2]$.

Contoh 2

Perhatikan pemanggilan berikut.

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

Di sini, satu-satunya subsekuens persekutuan dari A dan B adalah sekuens kosong $[]$. Maka dari itu, prosedur harus mengembalikan *array* kosong $[]$.

Contoh 3

Perhatikan pemanggilan berikut.

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

Di sini, subsekuens persekutuan dari A dan B adalah $[], [0], [1], [0, 1]$ dan $[1, 0]$. Dapat dibuktikan bahwa tidak ada subsekuens persekutuan semesta. Oleh karena itu, prosedur harus mengembalikan $[-1]$.

Contoh Grader

Format masukan:

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

Format keluaran:

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

Di sini, R adalah *array* yang dikembalikan ucs dan T adalah panjangnya.