

## Estudiando una civilización antigua

*Contribución de Lautaro Lasorsa*

### Descripción del problema

Un grupo de arqueólogos locales están estudiando el idioma de una civilización antigua. Para eso, utilizan  $M$  palabras presentes en textos que se han podido recuperar.

Saben que en este idioma existían letras vocales y consonantes, y que una regla muy importante del idioma era que en una misma palabra **no podían escribirse dos vocales o dos consonantes de forma consecutiva**. Eran válidas tanto las palabras que iniciaban en consonante, como las que iniciaban en vocal, y las palabras podían terminar en consonante o en vocal indistintamente.

Actualmente desean identificar cuáles letras del antiguo alfabeto eran vocales y cuáles eran consonantes. Como disponen de  $M$  palabras solamente, es posible que haya muchas formas diferentes de reconstruir cuáles letras eran vocales y cuáles consonantes.

A los arqueólogos les sirve cualquier reconstrucción tal que las reglas anteriores se cumplan para las  $M$  palabras que ya conocen. También están interesados en saber cuántas reconstrucciones válidas existen.

Los arqueólogos están completamente seguros de que las palabras descubiertas hasta ahora son correctas, por lo que siempre existirá al menos una asignación válida.

Se sabe que el alfabeto antiguo contiene exactamente  $N$  letras, que se identifican con enteros desde 1 hasta  $N$ .

### Detalles de implementación

Debes implementar la función `civilizacion(N, palabras, vocales, consonantes)`. Sus parámetros son:

- $N$ : un entero que indica la cantidad  $N$  de letras en el alfabeto
- `palabras`: un arreglo de  $M$  arreglos de enteros, que contiene las palabras descubiertas por los arqueólogos. `palabras[i]` es un arreglo de enteros que indica en orden las letras de la  $i$ -ésima palabra.
- `vocales`, `consonantes`: arreglos de enteros en los que se debe escribir una posible distribución de las letras como vocales y consonantes. En `vocales` se deben escribir los números correspondientes a las letras vocales, y en `consonantes` los de las consonantes. El orden en que se indiquen las letras dentro de cada uno de estos arreglos es indistinto.

Además, la función debe retornar un entero con la cantidad total de reconstrucciones válidas posibles. **Como esta cantidad puede ser muy grande, debes retornar el resto de dividirla por  $10^9+7 = 1000000007$**

### Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con  $N$  y  $M$
- $M$  líneas, cada una de las cuales describe una palabra. La  $i$ -ésima comienza con un entero  $K_i$  que indica la cantidad de letras en esa palabra, y luego contiene  $K_i$  enteros entre 1 y  $N$  correspondientes a cada una de las letras, en el orden en que aparecen en la palabra.

El evaluador escribe en la salida estándar una primera línea con el valor retornado por la llamada `civilizacion(N, palabras, vocales, consonantes)`, una segunda línea con el contenido del vector *vocales* y una tercera línea con el contenido del vector *consonantes*.

### Restricciones

- $1 \leq N, M \leq 200.000$
- $1 \leq K_i$ , y  $\sum_{i=1}^M K_i \leq 500.000$

### Ejemplos

Si el evaluador recibe:

```
9 1
9 1 2 3 4 5 6 7 8 9
```

Una solución correcta podría ser:

```
2
1 3 5 7 9
2 4 6 8
```

En cambio, si recibe:

```
10 10
1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
2 1 2
```

Podrá devolver:

```
512
1 3 4 5 6 7 8 9
2 10
```

Notar que puede haber letras que no aparezcan en ninguna de las palabras recuperadas por los arqueólogos.

### Puntuación

Se recibe el 50 % del puntaje por retornar el valor correcto. Se recibe el 50 % del puntaje por escribir una asignación válida en los arreglos *vocales* y *consonantes*.

### Subtareas

1.  $1 \leq N \leq 10, 1 \leq M \leq \sum_{i=1}^M K_i \leq 2.000$  (22 puntos)
2. Ninguna letra aparece en 2 o más palabras (14 puntos)
3. Hay una palabra que contiene todas las letras al menos una vez (14 puntos)
4. Sin restricciones adicionales (50 puntos)