

## 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 una serie de palabras presentes en textos que han podido recuperarse.

Además, saben que en esta civilización antigua también existían letras vocales y consonantes, y se sabe que una regla muy importante de su idioma era que en una palabra no podían escribirse dos vocales o dos consonantes de forma consecutiva. Sin embargo, una palabra puede iniciar o terminar tanto con una consonante como con una vocal.

Lo que desean saber actualmente es cuales letras del alfabeto de esta civilización son vocales y cuales son consonantes. Como las palabras de las que disponen son limitadas, es posible que haya más de una forma de reconstruir cuales letras eran vocales y cuales consonantes. Por ahora, cualquier reconstrucción que cumpla las reglas antes explicadas con las palabras que ya conocen les sirve.

Finalmente, por lo dicho anteriormente, están interesados en saber cuántas reconstrucciones validas posibles hay con las palabras descubiertas actualmente. Dado que este número puede ser muy grande desean saber el primer entero mayor o igual a su logaritmo en base 2.

Los arqueólogos están completamente seguros de que las palabras descubiertas hasta ahora son correctas, por lo cual no debes preocuparte por la posibilidad de que no se cumplan las reglas antes mencionadas.

Como el alfabeto utilizado por esta civilización era mucho más grande que el nuestro, los arqueólogos han optado por asignarle un número del 1 al  $N$  a cada una

de las letras.

### Detalles de implementación

Debes implementar la función `Civilizacion(N, palabras, vocales, consonantes)`, siendo  $N$  un ENTERO que indica la cantidad de letras en el alfabeto (están numeradas de 1 a  $N$ ), *palabras* un VECTOR DE VECTOR DE ENTERO que contiene las palabras descubiertas por los arqueólogos (*palabras<sub>i</sub>* es la  $i$ -ésima palabra), y *vocales* y *consonantes* son dos VECTOR DE ENTERO donde se debe devolver una posible distribución de las letras como vocales y consonantes.

Es decir, en el vector *vocales* deben aparecer los números correspondientes a las letras vocales, y lo análogo en el vector *consonantes*. En cada uno de estos vectores los enteros pueden estar en cualquier orden.

Además, la función debe retornar un ENTERO que sea el primer entero mayor o igual al logaritmo en base 2 de la cantidad de reconstrucciones validas posibles.

### Evaluador local

El evaluador local leera primero 2 enteros,  $N$  y  $M$ , la cantidad de letras en el alfabeto y la cantidad de palabras respectivamente.

Luego, leera  $M$  líneas, en la  $i$ -ésima de estas líneas leera primero un entero  $K_i$  que indica la cantidad de letras en esa palabra, y luego  $K_i$  enteros entre 1 y  $N$  correspondientes a cada una de las letras.

Posteriormente llamará a la función `Civilizacion(N, palabras, vocales, consonantes)`, y mostrará 3 líneas. La primera con el entero devuelto por la función, la tercera con el contenido del vector *vocales* y la tercera con el

contenido del vector *consonantes*.

### Cotas

- $1 \leq N, M \leq 250.000$
- $M \leq \sum_{i=1}^M K_i \leq 1.000.000$

### Ejemplos

Si el evaluador local recibe la siguiente entrada:

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

Una implementación correcta podrá devolver:

```
1
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:

```
9
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.

### 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)