TAD de Matrices de Toeplitz

Decimos que una matriz cuadrada es de *Toeplitz* si todos los elementos que se encuentran en una misma diagonal son iguales. Por ejemplo:

$$\begin{pmatrix} 4 & 9 & 1 & 0 \\ 0 & 4 & 9 & 1 \\ 3 & 0 & 4 & 9 \\ 7 & 3 & 0 & 4 \end{pmatrix}$$

Estas curiosas matrices deben su nombre al matemático alemán Otto Toeplitz, quien dedicó parte de su carrera a investigarlas. Entre las aplicaciones de este tipo de matrices está el desarrollo de algoritmos eficientes para el cálculo de la transformada de Fourier.

Este ejercicio consiste en implementar un TAD que almacene información sobre los elementos de una matriz de Toeplitz. Aunque la definición original de Toeplitz se aplica solamente a matrices cuadradas, vamos a extenderla para matrices rectangulares, es decir, para matrices en las que el número de filas es distinto del número de columnas. Por simplicidad, también supondremos que los elementos de la matriz son números enteros.

El TAD debe soportar tres operaciones:

- **Constructor**: recibe tres números enteros n, m y v (donde $n, m \geq 1$) y construye una matriz de Toeplitz de n filas y m columnas en la que todos los elementos tienen el valor v.
- **`get`**(i, j): suponiendo que `this` es una matriz $n \times m$ y que $0 \leq i < n \wedge 0 \leq j < m$, devuelve el elemento de la posición (i, j) de la matriz.
- **`set`**(i, j, v): suponiendo que `this` es una matriz $n \times m$ y que $0 \leq i < n \ \wedge \ 0 \leq j < m$, asigna el valor v a la posición (i, j) de la matriz y a todas las de su misma diagonal, para que la matriz resultante siga siendo de Toeplitz.

Se pide:

1. Implementar las operaciones del TAD de la manera más eficiente posible.
2. Indicar y justificar el coste asintótico en tiempo de cada una de las tres operaciones.
3. Escribir un programa que lea de la entrada la descripción de una secuencia de operaciones y las aplique a una instancia del TAD, llamando a los métodos correspondientes. El formato de la entrada y salida se describe a continuación.

Entrada

La entrada está formada por varios casos de prueba. Cada caso de prueba comienza por tres números n, m y v, que especifican las dimensiones de la matriz (n filas × m columnas) y el valor inicial v de todos los elementos de la misma. Las siguientes líneas describen la secuencia de operaciones que deben aplicarse a la matriz. Cada una de ellas comienza con la palabra `get` o `set`, seguida de los argumentos correspondientes. La secuencia de operaciones termina con la palabra `FIN`.

La entrada finaliza con un caso de prueba con tres ceros (`0 0 0`), que no debe procesarse.

Las dimensiones de cada matriz son números menores o iguales que 20 000. Los elementos de la matriz pueden ser números enteros comprendidos entre -10^9 y 10^9.

Salida

Para cada operación `get` debe imprimirse una línea con el valor devuelto por la misma.

Al final de cada caso de prueba debe imprimirse una línea con tres guiones (`---`).

Entrada de ejemplo

```
3 4 0
set 1 2 5
get 1 2
get 0 0
get 0 1
set 0 0 -2
get 1 1
FIN
1 6 -2
set 0 4 5
get 0 1
get 0 2
get 0 4
FIN
0 0 0
```

Salida de ejemplo

```
5
0
5
-2
---
-2
-2
5
---
```

Créditos

Autor: Manuel Montenegro

Imagen: Oberwolfach Photo Collection, archives of P. Roquette, Heidelberg (CC BY-SA 2.0)