

## C05: Matrices de Toeplitz

Estructuras de Datos  
Facultad de Informática - UCM

Este ejercicio consta de una única entrega, que debe enviarse al problema *DOMjudge* con identificador C05 antes del **jueves 5 de marzo a las 23:55**.

La entrega consiste en un único fichero `.cpp` que se subirá a *DOMjudge*. Puedes subir tantos intentos como quieras. Se tendrá en cuenta el último intento con el veredicto `CORRECT` que se haya realizado antes de la hora de entrega por parte de alguno de los miembros del grupo.

No olvides poner el nombre de los componentes del grupo en el fichero `.cpp`. Solo es necesario que uno de los componentes del grupo realice la entrega.

**Evaluación:** Este ejercicio se puntuará de 0 a 10. Para poder obtener una calificación superior a 0 es necesario obtener un veredicto `CORRECT`.

Una matriz de *Toeplitz* es aquella en cada diagonal descendente tiene sus elementos iguales. Por ejemplo, la siguiente matriz cumple esta propiedad:

$$\begin{pmatrix} 1 & 6 & 9 & 4 & 5 \\ 3 & 1 & 6 & 9 & 4 \\ 6 & 3 & 1 & 6 & 9 \\ 7 & 6 & 3 & 1 & 6 \end{pmatrix}$$

En este ejercicio deberás implementar un programa que determine cuáles de las matrices recibidas como parámetro son de Toeplitz. Para obligarte a utilizar iteradores, supondremos que las matrices recibida se representan mediante el tipo `list<list<int>>`, donde `std::list` es la clase de la STL que implementa el TAD Lista utilizando listas enlazadas.

Para realizar este ejercicio, realiza los siguientes pasos:

1. Implementa una función `es_toeplitz` que determine si la matriz pasada como parámetro es una matriz de Toeplitz:

```
bool es_toeplitz(const list<list<int>> &matriz)
```

La matriz recibida como parámetro se representa mediante una lista de listas. Las listas internas contienen los elementos de cada fila. Por ejemplo, la matriz mostrada arriba se representa mediante la siguiente lista de listas:

```
[[1, 6, 9, 4, 5], [3, 1, 6, 9, 4], [6, 3, 1, 6, 9], [7, 6, 3, 1, 6]]
```

A la hora de implementar esta función, puedes suponer que la matriz de entrada tiene, al menos, una fila. Además, puedes suponer que todas las listas internas tienen la misma longitud, y que esta longitud es también mayor o igual a uno.

**Importante:** Debes realizar esta función iterando directamente sobre las listas de la matriz de entrada. **No** está permitido hacer una copia de los elementos de `matriz` a otra estructura auxiliar (por ejemplo, un `std::vector` o un `array`).

2. Indica, mediante un comentario en el código fuente, el coste en tiempo de la función anterior. El coste debe estar expresado en función de las dimensiones de la matriz de entrada.

3. Escribe un programa que reciba de la entrada una serie de matrices e imprima, para cada una de ellas, la cadena SI o NO indicando si la matriz es de Toeplitz o no. El formato de la entrada y la salida se describen en las siguientes secciones.

## Entrada

La entrada contiene una serie de casos de prueba. Cada uno de ellos describe una matriz de  $n \times m$ , donde  $1 \leq n, m \leq 2000$ . Cada matriz se representa mediante  $n+1$  líneas. Las primeras  $n$  líneas contienen los  $m$  elementos de cada fila de la matriz, seguidos de un 0, que no forma parte de la matriz. La última línea contiene solamente un 0, que indica el fin de la matriz. Suponemos que los elementos de la matriz son números enteros comprendidos entre 1 y 100.

La entrada finaliza con un 0, que no se procesa.

## Salida

Para cada caso se escribirá una línea con la cadena SI si la matriz es de Toeplitz, o con la cadena NO en caso contrario.

## Entrada de ejemplo

```
3 2 1 0
4 3 2 0
5 4 3 0
0
8 6 1 6 5 0
9 8 6 1 6 0
3 9 8 6 1 0
7 3 9 8 6 0
0
2 1 0
3 2 0
4 3 0
6 5 0
0
0
```

## Salida de ejemplo

```
SI
SI
NO
```