Enunciado

Se dice que una matriz cuadrada, es decir que tiene el mismo número de filas que de columnas, es triangular cuando todos los valores que están por encima o por debajo de la diagonal principal son cero. También son triangulares aquellas matrices que cumplen estas dos condiciones a la vez.

$$T_3 = \begin{bmatrix} \mathbf{1} & 2 & 3 \\ 0 & \mathbf{6} & 4 \\ 0 & 0 & \mathbf{5} \end{bmatrix}$$

Realiza un programa que diga si una matriz cuadrada dada es o no triangular.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con un número que representa el número de filas, mayor que cero y menor o igual que 50, de la matriz cuadrada. A continuación se dan los elementos que forman la matriz. La entrada terminará con una matriz de 0 filas.

Salida

Para cada caso de prueba se indica a SI si la matriz es triangular y NO en caso contrario.

El presente ejercicio pertenece a la plataforma *AceptoElReto*.
[Matrices triangulares - ¡Acepta el reto!]
Autores: Patricia Dıaz Garcıa, Marco Antonio Gomez Martın y Pedro Pablo Gomez Martın.

Resolución

La presente resolución tiene como objetivo analizar y explicar en detalle la solución de un problema relacionado con la identificación de matrices cuadradas triangulares. Una matriz cuadrada se considera triangular cuando todos sus elementos por encima o por debajo de la diagonal principal son cero. La solución se proporciona mediante un programa implementado en el lenguaje de programación Python.

Descripción del Problema

Se plantea un escenario en el que se deben analizar varias matrices cuadradas para determinar si son o no triangulares. Cada matriz se representa como un vector de vectores, donde cada subvector contiene los elementos de una fila de la matriz. El problema consiste en verificar si todos los elementos por encima o por debajo de la diagonal principal de la matriz son iguales a cero. La entrada se compone de una serie de casos de prueba, donde cada caso inicia con el número de filas de la matriz y los elementos que componen la matriz. El programa debe imprimir "SI" si la matriz es triangular y "NO" si no lo es.

Utilizando el pensamiento computacional podemos dividir nuestro problema en subproblemas más pequeños.

- Subproblema 1: ¿Cómo definimos la matriz?

Tenemos como dato que las matrices que nos ingresan son cuadradas, por lo tanto, tienen la misma cantidad de filas que de columnas lo cual nos facilita el trabajo. Nuestras matrices van a ser entonces de 3x3, 4x4, … , y así sucesivamente.

- Subproblema 2: ¿Cómo se debe tomar el input para llenar nuestra matriz?

El enunciado nos muestra cómo se ingresan los inputs para formar nuestra matriz, se van ingresando de a filas.

1 2 3

Por lo tanto, podemos optar por caminos, el primero (*el que se encuentra presente en dicha resolución*) es tomar el input que es una cadena, quitarle los espacios y por medio de la operación subcadena ir completando nuestra matriz con los valores ingresados. La segunda alternativa es utilizando **una versión casera, es decir propia**, de la función split y guardando vectores en un vector y de esta manera generar nuestra matriz directamente cargada.

- Subproblema 3: ¿Cómo se cuando una matriz es triangular?

Si analizamos la cantidad de ceros que tiene una matriz triangular notaremos que está siempre va a ser múltiplo de 3, indistintamente del tamaño de la matriz (siempre y cuando sea cuadrada como en este caso). Entonces sabemos que como la cantidad es múltiplo de 3 si hacemos el módulo de esta siempre debe ser cero.

Implementación

La solución del problema se proporciona mediante un programa en Python que consta de dos funciones principales y un bucle de control. A continuación, se describen en detalle las partes clave de la implementación:

Función diagonalMatriz(matriz, longitud)

Esta función se encarga de determinar si una matriz cuadrada dada es triangular. La función acepta dos argumentos: “matriz”, que representa la matriz a ser evaluada, y “longitud”, que denota el número de filas (y columnas) de la matriz cuadrada. La función realiza dos comprobaciones separadas, una para la diagonal superior y otra para la diagonal inferior de la matriz. En cada comprobación, se utiliza un contador para contar la cantidad de elementos nulos (ceros) por encima o por debajo de la diagonal principal. La variable “contador” se inicializa en cero para cada comprobación.

- Diagonal Superior: se utiliza un bucle anidado “for” para iterar a través de los elementos de la matriz. La condición “i != k” garantiza que solo se consideren elementos fuera de la diagonal principal. Si “k > i” y el elemento “matriz[i][k]” es igual a cero, se incrementa el contador. Esta comprobación se realiza para los elementos por encima de la diagonal principal.

- Diagonal Inferior: Similar a la comprobación de la diagonal superior, pero verifica los elementos por debajo de la diagonal principal utilizando la condición si “k < i” y el elemento “matriz[i][k]” es igual a cero, se incrementa el contador.

Después de cada comprobación, la función verifica si el contador es un múltiplo de 3 y no es igual a cero. En caso afirmativo, la función devuelve "SI", indicando que la matriz es triangular. En caso contrario, devuelve "NO".

Función separarCadena(cadena)

Esta función se utiliza para eliminar los espacios en blanco de una cadena. En el contexto del problema, se aplica a cada fila de la matriz de entrada para separar los elementos numéricos.

Procesamiento de Entrada y Salida

El programa principal inicia con un bucle de control que se ejecuta mientras evalúa que la variable “entrada” sea diferente de cero. En cada iteración, se lee el número de filas de la matriz y se almacena en la variable “entrada”. Se verifica que “entrada” no sea igual a cero para continuar con la evaluación de la matriz.

- Se crea una matriz vacía “matriz” que se inicializa con ceros y tiene un tamaño de “entrada x entrada”. Los elementos de la matriz se leen línea por línea y se almacenan en esta estructura de datos.

- La función “separarCadena” se utiliza para eliminar los espacios en blanco de cada fila de la matriz de entrada.

- Se llama a la función “diagonalMatriz” para determinar si la matriz es triangular y el resultado se almacena en un vector llamado “resultados”.

- Finalmente, se incrementa la variable “a” para rastrear el número de casos de prueba procesados.

Salida de Resultados

Después de evaluar todas las matrices, el programa recorre la lista de resultados y los imprime. Solo se imprimen los resultados distintos de cero, lo que significa que solo se muestran "SI" o "NO" para los casos válidos.

Conclusion

En general, este programa ilustra la importancia de la programación eficiente para resolver problemas matemáticos y cómo se pueden utilizar estructuras de control y funciones en Python para abordar desafíos específicos.