

Problemas con submatrices

Nos puede interesar **ver una matriz como un conjunto de submatrices**. Muchas veces es un conjunto de submatrices que no se intersectan y que se obtienen al recorrer la matriz como si escribiéramos en un papel con bolígrafo (de izquierda a derecha y de arriba abajo). **Por ejemplo, tenemos una matriz de 11x11 y queremos ver sus submatrices de 3x3**

| | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|-------|
| 0-0 | 0-1 | 0-2 | 0-3 | 0-4 | 0-5 | 0-6 | 0-7 | 0-8 | 0-9 | 0-10 |
| 1-0 | 1-1 | 1-2 | 1-3 | 1-4 | 1-5 | 1-6 | 1-7 | 1-8 | 1-9 | 1-10 |
| 2-0 | 2-1 | 2-2 | 2-3 | 2-4 | 2-5 | 2-6 | 2-7 | 2-8 | 2-9 | 2-10 |
| 3-0 | 3-1 | 3-2 | 3-3 | 3-4 | 3-5 | 3-6 | 3-7 | 3-8 | 3-9 | 3-10 |
| 4-0 | 4-1 | 4-2 | 4-3 | 4-4 | 4-5 | 4-6 | 4-7 | 4-8 | 4-9 | 4-10 |
| 5-0 | 5-1 | 5-2 | 5-3 | 5-4 | 5-5 | 5-6 | 5-7 | 5-8 | 5-9 | 5-10 |
| 6-0 | 6-1 | 6-2 | 6-3 | 6-4 | 6-5 | 6-6 | 6-7 | 6-8 | 6-9 | 6-10 |
| 7-0 | 7-1 | 7-2 | 7-3 | 7-4 | 7-5 | 7-6 | 7-7 | 7-8 | 7-9 | 7-10 |
| 8-0 | 8-1 | 8-2 | 8-3 | 8-4 | 8-5 | 8-6 | 8-7 | 8-8 | 8-9 | 8-10 |
| 9-0 | 9-1 | 9-2 | 9-3 | 9-4 | 9-5 | 9-6 | 9-7 | 9-8 | 9-9 | 9-10 |
| 10-0 | 10-1 | 10-2 | 10-3 | 10-4 | 10-5 | 10-6 | 10-7 | 10-8 | 10-9 | 10-10 |

Tendremos 9 submatrices 3x3. En gris las zonas sin asignar a submatrices ya que 11 no es múltiplo de 3

Un código que nos permite obtener las coordenadas del origen (esquina superior izquierda) de las submatrices anteriores podría ser

```
public class Unidad4 {
    public static void main(String[] args) {
        String[][] matriz = new String[11][11];

        for (int i = 0; i < 11; i++) {
            for (int j = 0; j < 11; j++) {
                matriz[i][j] = i + "-" + j;
            }
        }
        System.out.println("Matriz original");
        for (String[] fila : matriz) {
            for (String s : fila) {
                System.out.print(s + "\t");
            }
            System.out.println("");
        }

        System.out.println("coordenadas origen Submatrices 3x3");
        for (int i = 0; i + 3 < matriz.length; i = i + 3) {
            for (int j = 0; j + 3 < matriz[i].length; j = j + 3) {
                System.out.print("(" + i + ", " + j + ") ");
            }
        }
        System.out.println("");

        System.out.println("otro razonamiento ");
        // habra mxn submatrices
        int m = matriz.length / 3;
        int n = matriz[0].length / 3;
```

```

int coordenadaFila = 0;
for (int i = 0; i < m; i++) {
    int coordenadaCol = 0;
    for (int j = 0; j < n; j++) {
        System.out.print("(" + coordenadaFila + "," + coordenadaCol + ") ");
        coordenadaCol += 3;
    }
    coordenadaFila += 3;
    System.out.println("");
}

System.out.println("otro razonamiento similar, hay muchos ....");
// franjasHorizontalesxfranjasVerticales es lo mismo que mxn
// pero cambiamos el nombre para distinguir solución
int franjasHorizontales = matriz.length / 3;
int franjasVerticales = matriz[0].length / 3;
for (int i = 0; i < franjasHorizontales; i++) {
    for (int j = 0; j < franjasVerticales; j++) {
        System.out.print("(" + i * 3 + "," + j * 3 + ") ");
    }
    System.out.println("");
}
}
}

```

Ejercicio U4_B4C_E1: Generalizamos un poco el código anterior escribiendo un método que nos imprima las coordenadas origen de las submatrices de cualquier rango $m \times n$ (no necesariamente $m=n$). Escribe el método para que funcione el código

```

public class Unidad4{
    static void obtenerOrigenSubMatrices(String[][] matriz,int tamFilas, int tamCol){
        //escribir código de este método
    }
    public static void main(String[] args) {
        String [][] matriz= new String[11][11];

        for (int i=0;i<11;i++){
            for(int j=0;j<11;j++){
                matriz[i][j]=i+"-"+j;
            }
        }
        System.out.println("Matriz original");
        for(String[] fila:matriz){
            for(String s:fila){
                System.out.print(s+"\t");
            }
            System.out.println("");
        }
        obtenerOrigenSubMatrices(matriz,5,5);
        obtenerOrigenSubMatrices(matriz,3,5);
        obtenerOrigenSubMatrices(matriz,5,3);
    }
}

```

Salida:

```

Matriz original
0-0    0-1    0-2    0-3    0-4    0-5    0-6    0-7    0-8    0-9    0-10
1-0    1-1    1-2    1-3    1-4    1-5    1-6    1-7    1-8    1-9    1-10
2-0    2-1    2-2    2-3    2-4    2-5    2-6    2-7    2-8    2-9    2-10
3-0    3-1    3-2    3-3    3-4    3-5    3-6    3-7    3-8    3-9    3-10
4-0    4-1    4-2    4-3    4-4    4-5    4-6    4-7    4-8    4-9    4-10
5-0    5-1    5-2    5-3    5-4    5-5    5-6    5-7    5-8    5-9    5-10
6-0    6-1    6-2    6-3    6-4    6-5    6-6    6-7    6-8    6-9    6-10
7-0    7-1    7-2    7-3    7-4    7-5    7-6    7-7    7-8    7-9    7-10
8-0    8-1    8-2    8-3    8-4    8-5    8-6    8-7    8-8    8-9    8-10
9-0    9-1    9-2    9-3    9-4    9-5    9-6    9-7    9-8    9-9    9-10
10-0   10-1   10-2   10-3   10-4   10-5   10-6   10-7   10-8   10-9   10-10

```

coordenadas origen Submatrices 5x5
(0,0) (0,5)
(5,0) (5,5)
coordenadas origen Submatrices 3x5
(0,0) (0,5)
(3,0) (3,5)
(6,0) (6,5)
coordenadas origen Submatrices 5x3
(0,0) (0,3) (0,6)
(5,0) (5,3) (5,6)

Ejercicio U4_B4C_E2: SUDOKUS CORRECTOS EN CODERUNNER