#include "modulo5.h"

#include <iostream>

Cerradura generarConfiguracionCerradura(const std::vector<int>& regla) {

int numEstructuras = (regla.size() / 5) + 1;

std::vector<int> tamanos(numEstructuras, 5); // Ejemplo: todas las estructuras de tamaño 5

Cerradura configuracion = configureCerradura(tamanos);

if (explorarConfiguraciones(configuracion, regla)) {

return configuracion;

} else {

std::cout << "No se encontró una configuración válida que cumpla con la regla." << std::endl;

return Cerradura(); // Devolver cerradura vacía

}

}

bool explorarConfiguraciones(Cerradura& configuracion, const std::vector<int>& regla) {

int numEstructuras = configuracion.estructuras.size();

int numRotacionesMaximas = configuracion.estructuras[0].filas; // Asumiendo matrices cuadradas

// Explorar todas las combinaciones de rotaciones

for (int i = 0; i < numRotacionesMaximas; i++) {

for (int j = 0; j < numRotacionesMaximas; j++) {

// ... (más anidaciones de bucles para las rotaciones restantes)

// Rotar las estructuras M según la combinación actual

rotarAntihorario(configuracion.estructuras[0], i);

rotarAntihorario(configuracion.estructuras[1], j);

// ... (rotar las estructuras restantes)

// Verificar si la configuración cumple con la regla

if (validarRegla(configuracion, regla)) {

return true; // Se encontró una configuración válida

}

// Restaurar rotaciones para la siguiente combinación

rotarAntihorario(configuracion.estructuras[1], numRotacionesMaximas - j);

rotarAntihorario(configuracion.estructuras[0], numRotacionesMaximas - i);

// ... (restaurar rotaciones de las estructuras restantes)

}

}

return false; // No se encontró una configuración válida

}