Imagen que contiene Patrón de fondo

Descripción generada automáticamente Dibujo con letras blancas

Descripción generada automáticamente con confianza media

**Universidad de Antioquia**

Facultad de Ingeniería

Programas educativos:

**Ingeniería de Telecomunicaciones**

**Ingeniería Electrónica**

Docentes:

**ANÍBAL GUERRA**

**AUGUSTO SALAZAR**

Parcial 1: Informática II

**Desafío 1**

**Informe para Desafío 1**

Estudiantes:

**ANTONIO CARLOS MERLANO RICARDO (1010152199)**

**JUAN ESTEBAN LOPEZ CASTRILLON (1127941113)**

**08 DE ABRIL DEL 2024**

**Semestre 2024 – 1**

**Análisis del Sistema de Cerraduras X de Informa2**

El sistema de cerraduras X desarrollado por Informa2 es una solución de seguridad que se basa en una estructura de datos especial, denominada M, capaz de rotar y cambiar de dimensiones. Esta estructura se caracteriza por ser cuadrada y tener un número impar de filas, lo que facilita su alineación central.

Estructura de datos M y Cerradura X:

La cerradura X se compone de varias estructuras M alineadas, utilizando la celda central como referencia. No hay restricciones en cuanto a la cantidad de estructuras M que se pueden alinear ni en cuanto a sus tamaños, lo que permite una flexibilidad en el diseño de las cerraduras. Por ejemplo, una cerradura X (5, 7, 5, 9) podría estar compuesta por cuatro estructuras de tamaños 5x5, 7x7, 5x5 y 9x9 respectivamente.

Regla de Validación K:

Para abrir una cerradura X, se debe validar la rotación de las estructuras M alineadas mediante una regla K específica. Esta regla K consiste en un conjunto de condiciones que deben cumplirse para que la cerradura se abra correctamente. Por ejemplo, la regla K (4,3,1,-1,1) indica que, para la celda ubicada en la fila 4, columna 3 de la primera estructura, su valor debe ser mayor al de la celda correspondiente en la siguiente estructura, y así sucesivamente.

Proceso de Apertura:

El proceso de apertura de la cerradura implica rotar cada una de las estructuras M de forma independiente, alineando las celdas de manera que se cumplan las condiciones establecidas por la regla K. Para ello, es necesario diseñar algoritmos capaces de crear y rotar matrices bidimensionales de manera eficiente, teniendo en cuenta la relación entre filas y columnas durante la rotación.

Pasos para Resolver el Problema:

1. Diseñar un algoritmo para crear arreglos bidimensionales con tamaño variable, utilizando arreglos dinámicos para evitar desbordamientos de memoria.
2. Implementar un algoritmo para rotar las estructuras M, teniendo en cuenta que, al rotar 90 grados, las filas se convierten en columnas y las columnas se invierten en orden.
3. Desarrollar un algoritmo para generar la cerradura X, que pregunte al usuario la cantidad y tamaño de las estructuras M que compondrán la cerradura, y cree las estructuras de manera adecuada.
4. Crear un algoritmo de validación que verifique si se cumplen las condiciones de la regla K para cada celda de la cerradura X, rotando las estructuras M según sea necesario hasta que la cerradura se abra.

NOTA: CONSIDERACIONES PARA LA ALTERNATIVA DE SOLUCIÓN PROPUESTA

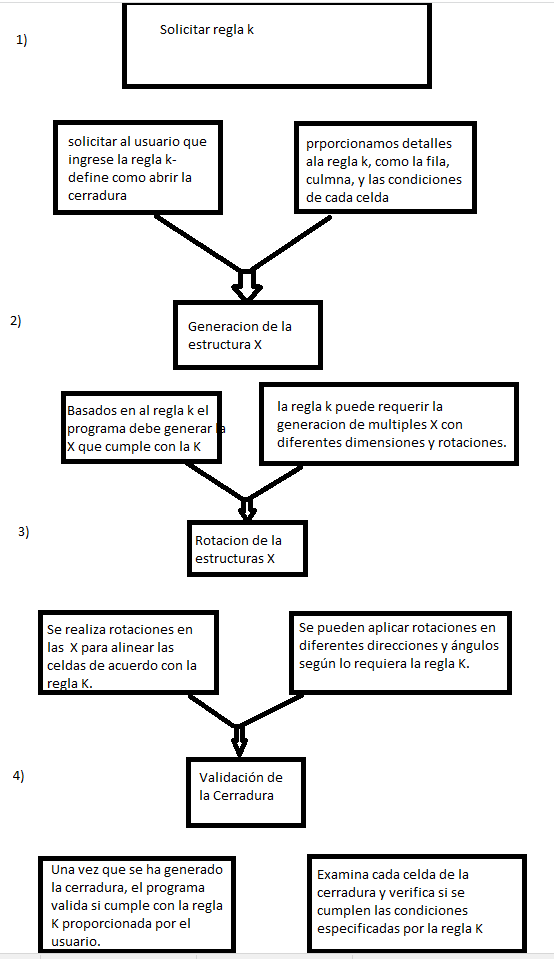
Es muy importante asegurar la eficiencia del algoritmo tanto en tiempo como en memoria, especialmente al diseñar las rutinas de generación y rotación de estructuras M. Esto implica implementar estrategias de afrontamiento óptimas, optimizadas para matrices de diferentes tamaños, evitando sobrecargas innecesarias. De manera similar, la regla K debemos verificar la para garantizar que las condiciones especificadas se cumplan adecuadamente en todas las celdas de la estructura M y la cerradura X.

Es fundamental realizar pruebas para verificar la funcionalidad y confiabilidad del sistema, probar diferentes configuraciones de la cerradura X e incorporar casos extremos para garantizar que el sistema funcione.

**Esquema implementado para el desarrollo del problema**

Para implementar la serie de pasos que utilizamos para el desarrollo del desafío primeramente fue dividir el programa por etapas, es decir:

Para cada etapa del desarrollo, utilizamos guiados por la guía donde una vez implementados los pasos ya analizados, para esto realizamos una especie de esquema donde este nos permita la comprensión del desarrollo en el programa:



**Algoritmos implementados**

**Problemas en el desarrollo:**

A medida que íbamos desarrollando el programa nos encontramos con varios problemas principalmente era entender y comprender el análisis ya previamente realizado para lograr plasmar la ideas en nuestro programa como por ejemplo el proceso mas complejo es realizar la validación de la regla K ya que esto representa el 60% de nuestro programa, por otro parte tuvimos varios problemas de sintaxis esto dado por algunos conceptos en memoria dinámica, la utilización de punteros pero que a medida íbamos utilizando mas el programa era más fácil de detectar por medio del uso dela depuración que nos simplifico en detectar los errores por ejemplo para el desarrollo de las matrices porque nos permite ubicarnos exactamente donde queremos implementar o modificar parte del código, por ultimo no solo es analizar y entender el desafío planteado si no como plasmar nuestras ideas para la realización del código, por otra parte toco aprender a utilizar correctamente la plataforma git hub y como trabajar en proyectos grupales y la realización y actualización de commit.

**EVOLUCION DEL PROGRAMA**

Para el desarrollo en qt dividimos el proyecto en varios archivos el main.cpp, funciones.cpp y función.h como se sobre entiende en el main implementaremos las funciones que desarrollamos en fucnione.cpp y en función.h están declaradas cada función

En lo que se desarrollo el programa se llevo paso a paso por la guía, a continuación, la evolución del programa por serie de pasos:

1. Definición de funciones: primero fueron estas funciones:
   * + - * Void pedirDatos()
         * Void crearEstructuras()
         * Void llenarEsctuctura()

Estas funciones se utilizaron para la creación de la estructura M es decir para la creación de matriz de tamaño 3x3 en adelante.

1. Implementación de las funciones para rotar dimensión de las matrices

* void girarIzquierda

1. Empezamos a utilizar las distintas reglas para K:

* void pedirClavek()
* void validarReglaK()

Estas funciones nos permiten que el usuario ajuste el numero de estructuras filas y columnas para validar la regla k

1. Validamos para generar la primera matriz de la cerradura: por medio de
2. En este paso nos enfocamos a la aplicación de la validación de la regla k por lo cual hicimos cambios e implementamos y borramos varias funciones