

Informe Desarrollo Del Proyecto

Nombres: -Alejandro Zapata Quintero

-Juana María Márquez Guzmán

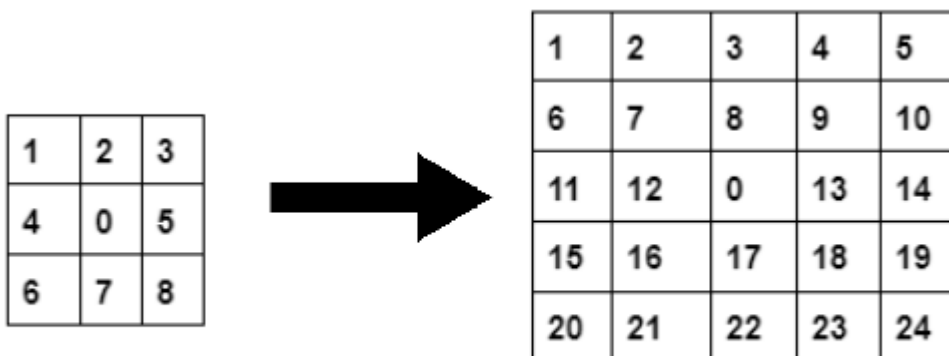
1. Análisis del problema y consideraciones para la alternativa de solución propuesta.

El problema presentado es un desafío de programación que implica el diseño de un sistema de seguridad para la empresa Informa2. Este sistema se basa en el concepto de cerraduras, denominadas X, que tienen características y requisitos específicos. Las cerraduras X están compuestas por varias estructuras M alineadas una tras otra. No hay restricción en la cantidad de estas estructuras y sus tamaños pueden variar.

El sistema de apertura de X funciona a partir de la validación de una regla K. Esta regla considera el valor de una celda específica, su posición y la ubicación dentro de las diferentes estructuras alineadas. Para abrir la cerradura, se deben rotar cada una de las estructuras de forma independiente. El objetivo es alinear las celdas de tal manera que la validación de K sea verdadera.

Para resolver este problema, se propuso una solución que implica la creación de una serie de funciones en C + +. Estas funciones permiten:

Crear estructuras de datos de tamaño variable: La primera tarea fue diseñar una estructura de datos que pudiera representar las estructuras M de la cerradura X. Estas estructuras M son matrices cuadradas de tamaño impar variable. Para representarlas, se decidió utilizar un arreglo de enteros en C + +. Esta estructura de datos permite crear matrices de cualquier tamaño y acceder a sus elementos de manera eficiente.



Realizar las rotaciones a las estructuras: Una vez que se tuvo una representación para las estructuras M, el siguiente paso fue implementar una función para rotar estas estructuras. La rotación es una operación clave en el funcionamiento de la cerradura X, ya que es la forma en que se alinean las celdas para validar la regla K. La función de rotación toma una estructura M como entrada y devuelve una nueva estructura M que es el resultado de rotar la original 90 grados en sentido antihorario.

1	2	3
4	0	5
6	7	8

Estado neutro

3	5	8
2	0	7
1	4	6

estado 1

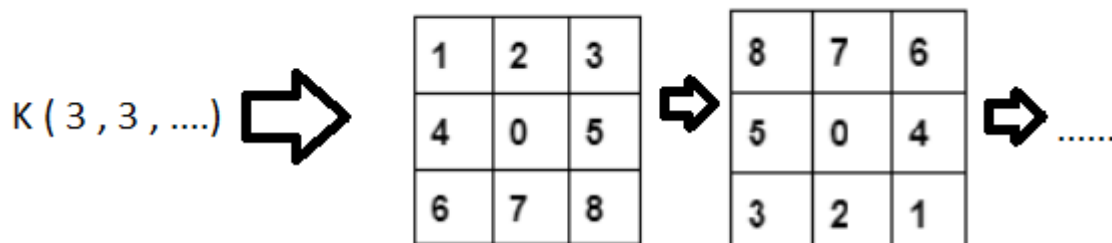
8	7	6
5	0	4
3	2	1

estado 2

6	4	1
7	0	2
8	5	3

estado 3

Configurar cerraduras de tal forma que la cantidad y el tamaño de las estructuras que la componen sea variable: Con las funciones para crear y rotar estructuras M, se pudo implementar una función para configurar una cerradura X. Esta función toma como entrada una lista de tamaños y crea una cerradura X con una cantidad de estructuras M igual a la longitud de la lista. Cada estructura M tiene un tamaño igual al elemento correspondiente de la lista.



Ejemplo: Variación de llave y cerradura.

Validar una regla de apertura sobre una cerradura: La siguiente tarea fue implementar una función para validar una regla K sobre una cerradura X. Esta función toma como entrada una cerradura X y una regla K, y devuelve un valor booleano que indica si la regla es válida para la cerradura. Para hacer esto, la función recorre las estructuras M de la cerradura y compara los valores de las celdas especificadas por la regla K.

Generar al menos una configuración de cerradura que se pueda abrir con una regla dada: Finalmente, se implementó una función para generar una configuración de cerradura que se pueda abrir con una regla K dada. Esta función toma como entrada una regla K y

devuelve una cerradura X que se puede abrir con esa regla. Para hacer esto, la función crea una cerradura X con una cantidad de estructuras M igual a la longitud de la regla K, y luego rota las estructuras hasta que la regla K sea válida.

Algunos pasos secundarios son:

Validación de entradas: Antes de procesar las estructuras M y las reglas K, es importante validar las entradas para asegurarse de que son válidas. Esto incluye comprobar que los tamaños de las estructuras M son positivos y que las reglas K se refieren a posiciones válidas dentro de las estructuras.

Manejo de errores: Durante la ejecución de las funciones, pueden ocurrir varios tipos de errores, como desbordamientos de memoria o violaciones de segmento. Es importante manejar estos errores de manera adecuada para evitar que el programa se bloquee y para proporcionar mensajes de error útiles al usuario..

Línea de desarrollo y ejecución

A la hora ya de ejecutar todo este desafío notamos que a la hora de plantear la solución habría que tomar ciertas condiciones y tener claros ciertos conceptos para que el desafío de programación pueda ser abordado con mayor precisión, las soluciones a los distintos problemas que fueron surgiendo son:

se nos dará una llave tal que $K(f, c, n, \dots, n)$, donde la llave tendrá obligatoriamente en sus primeros 2 elementos la posición en fila y columnas respectivamente f y c, esta posición será ubicada en la primera estructura M (Matriz), luego se le ingresaran valores n que pueden tomar valores de -1, 0 y 1, estos valores se podrán ingresar sin límite es decir la llave podría variar su tamaño y tener tanto 5 como 10 elementos, Las soluciones que tomamos en este apartado son:

- La llave tendrá que tener por lo menos 3 valores para que funciones el programa, estos valores son fila, columna y por lo menos una comparación

- Para dar el primer valor de la cerradura y de los estados de la estructura M, es decir las rotaciones, se establecieron 2 reglas útiles para un mejor desarrollo, y es que a la hora que el usuario de una llave K, por ejemplo $K(5,5,\dots)$ podremos notar que esta ubicación se refiere a la fila 5 y columna 5 por lo que la primera matriz tendrá que ser por lo menos de tamaño 5×5 , entonces esta regla servirá tomando el mayor valor impar de estas coordenadas, y en caso de ser par será el mayor valor par + 1, tal que si nos dan por ejemplo $K(2,6,\dots)$ la estructura M de menor tamaño que podrá contener esta llave será de tamaño 7×7 , entonces el programa que haremos no tendrá que procesar matrices con menor tamaño al posible ya que en este caso no tiene sentido que después de este arreglo halla una matriz en la cerradura de menor tamaño que 7×7 y como este primer valor no tiene que cumplir ninguna comparación estará en estado neutro por definición.

-teniendo como base la menor estructura y pensando que estas matrices están alineadas desde el centro , se puede ver que para hallar las distintas comparaciones además de rotar se tendrá que aumentar la dimensión de las matrices , notamos que entre cada 2 dimensiones que suba una matriz , la posición inicial subirá 1 en fila y 1 en columna , visto desde un ejemplo si nos dan una llave cuyos primeros 2 valores sean $K(3,2,...)$ notamos que tiene una posición 3,2 en una matriz 3×3 , sin embargo en la matriz 5×5 esa posición aumentaron 1 en columna y 1 en suma es decir sería igual a 4,3, y así sucesivamente para las distintas matrices.

