Para resolver el problema vamos a partirlo en varios módulos, un módulo que se encargue de generar las matrices.

Otro módulo que se encargue de crear una lista de los elementos de las matrices generadas de orden n en la celda determinada por la llave.

Otro que se encargue de ordenar dicha lista de acuerdo con las condiciones de la llave, generando una lista encadenada con los valores, posición

Dentro del módulo de crear lista, habrá de desarrollarse un módulo que calcule el valor de la posición relativa de la siguiente matriz respecto a la actual.

Hay que tener en cuenta que las matrices han de tener orden impar, comenzando en 3, esto para que se pueda generar una matriz con celda eje vacía, de manera tal que al rotarla siga permaneciendo “vacía” dicha celda.

El tamaño inicial (ó mínimo) de la matriz inicial ha de estar determinado por los valores de la clave, donde el valor mayor ya sea de la fila o columna nos va a ayudar a determinar el orden de la matriz a generar. Así, si este valor esta entre 1 y 3 vamos a iniciar con una matriz cuadrada de orden 3, si es un valor entre 4 y 5 inclusives, vamos a iniciar generando una de orden 5, entre 6 y 7 con una de orden 7, y asi sucesivamente.

EJEMPLO

Para facilitar el análisis del problema vamos a ver casos extremos donde vemos una llave que tiene el valor K(1,2: 1 1 1 1)

|  |  |  |
| --- | --- | --- |
| 8 | 7 | 6 |
| x |  | x |
| x | x | x |

Estado S2 (segunda Rotación)

|  |  |  |
| --- | --- | --- |
| 3 | 5 | 8 |
| x |  | x |
| x | x | x |

Estado S1 (primera Rotación)

|  |  |  |
| --- | --- | --- |
| 6 | 4 | 1 |
|  |  |  |
| x | x | x |

Estado S3 (tercera y última rotación, antes de volver a la posición original o neutra)

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| x |  | x |
| x | x | x |

Posición neutra S0 (sin rotación).

Este ejemplo nos sirve para ilustrar que después de tres “giros” la estructura vuelve a su estado original. Sin embargo, lo que nos interesa en esta parte es notar que ya no quedan rotaciones posibles para obtener un número menor que 2.

Por lo tanto, deberemos recurrir a una matriz de orden superior para intentar obtener otro número, la siguiente matriz será de orden 5,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 |  | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 |

Tomaremos de la posición virtual 1,2 el valor 8.

Así obtenemos la lista 7, 5, 4, 2 y 8.

Recordemos los valores de validación de la clave: 1, 1, 1 y 1.

Ya sabíamos que hasta el 2 se cumplía el tercer valor de validación, pero veamos si se cumple el cuarto.

Como el cuarto valor de validación es un uno, esto nos dice que el cuarto valor de la lista ha de ser mayor que el siguiente: ¿Es 2 mayor que 8? No. Por lo tanto, el 8 ha de pasar a la izquierda del 2, luego volvemos a preguntar si es mayor, si es así proseguimos, si no es porque es igual valor, por lo tanto, debemos obtener otro número girando al estado S1 la matriz de orden 5.

Como 8 es mayor que 2, entonces la lista quedará así 7, 5, 4, 8 y 2. Hasta ahí pareciera que ya hemos encontrado los valores de las matrices que satisfacen la combinación de la llave, pero no, ahora deberemos comprobar la tercera condición: 7 5 ( 4 8 ) y 2.

Es 4 mayor que 8, no, entonces el 8 pasa a la izquierda, del 4: 7, 5, 8, 4 y 2.

Ahora agrupamos el 5 y el 8 que se corresponden con la segunda condición: 7 ( 5 8) 4 y 2.

Hacemos la pregunta correspondiente, y así sucesivamente llegamos a la lista enlazada

8 7 5 4 2.

Tener estas listas de valores no nos va a dar información necesaria para luego recuperar las matrices con sus respectivas orientaciones (rotaciones), por lo que se hace necesario crear una lista enlazada con varios datos, como lo son el valor como tal de la celda en dicha matriz, el orden y el estado de esta.

Ahora nos vamos a centrar en una lista de 4 numeros para ordenarlos de acuerdo a las condiciones de la llave. Asi que teniendo una lista de 4 numeros L:(2, 5, 7, 4) vamos a tratar de cumplir esta cadena +1 –1 0 +1:

Bajamos el 2(izq,) y lo comparamos con el 2(der):

Es el 2 de la izquierda mayor al de la derecha? Como la respuesta es no, entonces hay dos opciones, que sean iguales, en dicho caso debe salirse y preguntar con el siguiente numero, y en caso de que sea menor intercambiar de posiciones.

Como el 2 es igual al dos, tiene que pasar a probar con el siguiente numero de la lista que es el 5.

¿2 es mayor que 5? No, entonces el 5 pasa a la izquierda del 2.

Volvemos a bajar el 2, queda 5(22), -1: ¿Es el 2 menor que el 2, como la respuesta es no, entonces hay dos opciones, una que sea mayor, y la otra es que sea igual. Como es igual, se evita, y se baja el siguiente numero: 5.

5(25) -1: Es el 2 menor que el 5, si, entonces la lista queda 525.

Ahora viene el cero que significa que el siguiente es igual. Comenzamos de nuevo bajando el 2: 52(52) como el 5 no es igual al 2, entonces bajamos el siguiente de la lista 52(55), como es igual, la lista queda 5255.

Ahora viene el +1, que significa que el ultimo es superior al siguiente, bajemos el 2, nuevamente, 525(52), ahora nos preguntamos si el 5>2, como lo es, la lista queda 52552.

Este analisis es un acercamiento al planteamiento de la solucion, sin embargo tendremos que abordar el problema que surge cuando se llega a un minimo o un máximo, y no habrá más numeros que colocar a la lista, en ese caso tocará recurrir a generar una matriz de mayor dimensión.

Lo que hay que tener en cuenta es que las matrices independiente de la dimension que tengan, siempre tendrán cuatro estados de acuerdo con su rotación. Esto nos servirá mas luego para plantear una solución para agregar cuatro valores a la lista de candidatos a satisfacer las condiciones de la llave, por ahora miremos un diagrama de flujo del algoritmo que nos permite organizar los valores de acuerdo a las condiciones.

Para facilitar el desarrollo de la solución, vamos a partir de una RESTRICCIÓN que consiste en que, para crear la lista de valores sacados de las matrices generadas, no se usaran nuevamente los ya encontrados en matrices inferiores. Ejemplo: si ya hemos encontrado el 14, el 23, el 17 y el 28, y en otra matriz el 15, el 32, el 14, y el 18, en la lista generada aparecerá el 14 una sola vez.

Teniendo esto como premisa vamos a ordenar ascendentemente los valores seleccionados.

Cuando aparezca un cero en la clave, omitiremos esta condición, ya que luego se podrá insertar en la lista organizada. O sea que solamente vamos a tener en cuenta la condición de cuando el valor de la siguiente matriz es diferente al valor de la actual. Se debe crear un módulo para determinar el valor en la posición relativa de las siguientes matrices a la inicial.

Ya con la lista ordenada ascendentemente, notaremos que los números, sin importar su valor, les corresponderá una posición en la lista de acuerdo con el peso que tengan. Esto quiere decir que para el siguiente análisis del problema podemos obviar los valores y simplemente ORGANIZARLOS de acuerdo con la posición en que aparezca. Por ejemplo, si los valores hallados son 9, 7 y 12. y las instrucciones de la clave corresponden a –1 y +1, después de ordenados ascendentemente quedarán 7, 9 y 12, y para hallar el orden en que deberán aparecer, nos referiremos a ellos como el primero, el segundo y el tercer valor, o mejor, con índices como el 1, el 2 y el 3.

Ilustremos mediante un ejemplo el diseño del algoritmo para hallar el orden en que deben ponerse las matrices halladas. Halle los posibles valores de las combinaciones de clave de apertura para tres valores organizados ascendentemente.

Para solucionar esto nos referiremos a ellos como 1, 2 y 3. Donde 1 se refiere al menor y 3 al mayor.

|  |  |  |  |
| --- | --- | --- | --- |
| - | 12 |  |  |
| + | 12 | 21 |  |
| -- | 1(23) | 123 |  |
| -+ | 1(23) | (13)2 | 132 |
| +- | 2(13) |  |  |
| ++ | 2(13) | (23)1 | 321 |

Explicamos la tabla, para construir un –1 en la clave, simplemente bajamos el 1 y a continuación el 2, como el número que bajemos siempre va a ser mayor a los existentes no toca hacer mas nada.

Para el +1, bajamos el 1. A continuación, bajamos el 2, como no se cumple 1>2, simplemente corremos el 2 una posición a la izquierda.

Para el –1,-1, ya teníamos construido el primer - (izquierda). Por lo que bajamos el resultado del -: 12. A continuación bajamos el 3. Vemos a ver si se cumple, -1 para la pareja (23), como se cumple, no se hace ningun cambio de posición: 123. Ratificamos que cuando haya un –1 en el “nuevo” condicionamiento, bastara con dejar el número intacto.

Para el –1 +1. Bajamos el resultado del –1 que corresponde a 12, luego agregamos a este el 3 al final. Comparamos el penúltimo con el último: 1(23), como no se cumple, procedemos a correr el 3 una posición a la izquierda. 132. Ahora comparamos la siguiente dupla: (13)2, ahi la condición es 1<3, como se cumple se queda así.

Para el +1 –1 bajamos el resultado de + que es 21. Luego agregamos el 3: 213. Agrupamos los dos últimos. 2(13), como sabemos que si el ultimo es un –1, no habrá que hacer nada, entonces dejamos el número 213.

Para el +1 +1, bajamos el resultado de +1 que es 21. Agregamos el 3: 213. Comparamos los dos últimos: 2(13), como no se cumple 1>3, entonces corremos el 3 una posición a la izquierda: 231. Comparamos las posiciones anteriores: (23)1. Ahora vemos que según la instrucción +1, la condición no se cumple, entonces corremos el 3 una posición a la izquierda. Es necesario resaltar que la condición previa no es necesario volverla a validar: miremos esto con la ayuda de una tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rab | Rbc | Rcd | Rde | Ref | 123456 |
| Rab | Rbc | Rcd | Rdf | Rfe | 123465 |
| Rab | Rbc | Rcf | Rfd | Rde | 123645 |
| Rab | Rbf | Rfc | Rcd | Rde | 126345 |
| Raf | Rfb | Rbc | Rcd | Rde | 162345 |
| Rfa | Rab | Rbc | Rcd | Rde | 612345 |

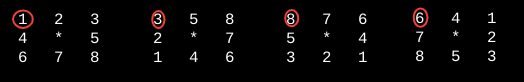
Consideremos un conjunto de relaciones entre duplas de 6 valores, previamente ordenadas 123456. Las partes con texto color rojo se refieren a las nuevas relaciones al mover el último número de posición, mientras que las partes resaltadas se refiere a que las relaciones serán las mismas entre los demás números a la derecha.

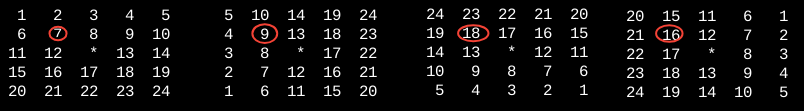
Se puede demostrar que para generar una organización bastara con saber cuántos más hay a la izquierda del último menos que aparezca, y esa misma cantidad correr el ultimo número tantas posiciones a la izquierda.

Por ejemplo, para la combinación --++ corresponde el orden 12543, entonces para el (--++)+ bastara con bajar el 6 y correrlo tres posiciones a la izquierda, ya que hay 3 + antes del -. 12543(6), quedando 12 6 543. Se puede notar que se cumple que 1<2, que 2<6, que 6>5,...

Si habiendo hallado --++, nos toca hallar un-, o sea (--++)- bastará con bajar el siguiente número: por ejemplo: 12543 es el orden correspondiente a --++, entonces (12543)6 será el correspondiente a (--++)-.

PROBLEMA CON LA MATRIZ INICIAL





Supongamos la clave K (1,1,+1,+1,+1,+1,+1,+1)

Entonces el algoritmo seleccionará los primeros 6 valores encontrados en la posición relativa 1,1. Estos corresponden a 1 3 8 6 7 y 9. Luego el módulo siguiente arrojara un orden para cumplir con la clave: 9 8 7 6 3 1. Si probamos las condiciones vemos que se cumplen las condiciones para la clave: +1: 9>8, +1: 8>7, +1: 7>6, +1: 6>3 y +1: 3>1. El problema está en que la clave dice que la posición absoluta de la primera clave es la 1,1, y resulta que esa no es la posición absoluta del nueve, ya que en realidad se trata de la posición 2,2.

¿Cómo se resuelve esto? Simplemente se vuelve a ejecutar el módulo, pero esta vez se comienza a generar las matrices desde un orden superior al actual. Por ejemplo, en este caso fue de orden 3, entonces empezamos a generar las matrices desde el orden 5.

En las matrices de orden cinco generamos 1,5,24 y 20. En las matrices de orden siete generamos el 9 y el 13.

El algoritmo generará la organización 24 20 13 ... El caso es que 24 pertenece a la matriz de menor tamaño, por lo tanto, esta será la respuesta adecuada. Si no se llegase hallar solución habría que seguir intentando, incrementando en 2 el orden de la matriz inicial, y así sucesivamente hasta dar con los valores que sí satisfagan la clave.

ALGORITMO GENERAL

El algoritmo general consta unos pasos secuenciales:

1. Ingreso de clave
2. Exclusión de ceros

Previo al paso tres se mira la posición en la clave, para ver con que orden(tamaño) de matriz comenzar.

1. Generación de matrices de acuerdo longitud de clave.
2. Generación de arreglo con valores en posición de acuerdo con la clave
3. Ordenamiento Ascendente de los valores seleccionados.
4. Generación de Orden de los valores del arreglo
5. Ordenamiento de la lista
6. Verificación de que el primer elemento pertenezca a la matriz de orden inferior.

Si es congruente el orden de la primera matriz con el orden de la matriz de orden inferior, se incrementa el orden en 2 y se regresa al paso 3.

1. Ordenamiento incluyendo ceros
2. Impresión de los órdenes(tamaño) de cada matriz y cantidad de giros a la izquierda para cada una.

