ISIS 1105 – 2020-2

Problema A – Inversa Matricial en Z_p

0. **Identificación**
    - Julián Oliveros – je.oliverosf – 201821595
    - Camilo Rozo – ce.rozob – 201820147
1. **Algoritmo de Solución**
    - Explicación del algoritmo elegido. Si hubo alternativas de implantación diferentes, explicar por qué se escogió la que se implementó.

Explicación Algoritmo elegido

Para el problema de Dada una matriz (M_n) y un número primo (Z_p), calcular la matriz inversa multiplicativa correspondiente, si ésta existe. Se tomo la decisión de implementar un algoritmo de Gauss con unas variaciones en cómo se realizaban la operaciones matemáticas, pues para la división o multiplicación se implemento el teorema de Fermat $a * x \; mod \; p = 1$ donde a es un número que pertenece a un celda de la matriz, p un número primo y x un número natural tal que cumpla la condición. Además, para remplazar la suma y la resta se utilizo la siguiente ecuación $R_2 = ((R_1 * (r) + R_2) mod \; p)$, esta operación es equivalente la operación en el método Gauss de $f_2 = f_2 + r * f_1$, Dónde R2 es la casilla a la que quiere convertir R1 es la fila donde se encuentra el pivote R es un número tal que vuelve 0 a R2 y p es un número primo. Estas fueron las únicas variaciones que se realizaron frente a el algoritmo de Eliminación de Gauss-Jordan.

En un principio se tomó la decisión de realizar un doble recorrido para recorrer las dos matrices (la matriz que nos pasan por parámetro y la matriz inversa), sin embargo pudimos analizar que solo eran más efectivo iterar sobre la diagonal de la matriz, pues es donde se quieren convertir estos valores en 1 para formar pivotes y a mediada que uno iba avanzando las columnas posteriores ya no eran relevantes, por lo que se fue iterando en la diagonal de la matriz convirtiendo estas casillas en pivotes, para después convertir los demás valores de la columna en 0 y realizar sus respectivos cambios a las celdas posteriores.

Se utlizaron unos metodos auxiliares para poder desarrollar el problema:

- **cambioDeFilas**: Este método se utliza cuando se llega una celda i en la diagonal de la matriz y esta es 0, pues lo que hace es buscar otra fila posterior con la que pueda cambiar de posición y si la encuentra realiza el cambio.

- **encontrarDivisorPor1:** Este método sirve para la ecuación $a * x \; mod \; p = 1$, pues su función es encontrar un x tal que se satisfaga la ecuación.

E/S	Nombre	Tipo	Descripción
E	p	$p \; \epsilon \; P$	Es número primo el cual se utiliza para realizar todas las funciones módulo.
E	n	nat	Es un número natural positivo que indica el tamaño de la matriz.

E	m	Array[0,N)[0,N)	Matriz de números entre 0 y p-1 de tamaño NxN.
S	MI	Array[0,N)[0,N)	Matriz inversa multiplicativa correspondiente de la matriz de entrada, en caso de que exista.

Precondición: $\{p \in P \ \wedge (\forall x \mid x \in m: 0 \ \leq x \geq p-1 \ )\}$

PostCondición:

Queremos que la matriz m original multiplicada por la solución encontrada módulo p sea la matriz identidad, de modo que definimos formalmente un predicado para saber si una matriz es la matriz identidad respecto a un módulo a

$$IDENTIDAD(C,a) \equiv (\forall x,y|0 \leq x,y \leq N \wedge x \neq y: C[x][y] \equiv_a 0) \wedge (\forall x|0 \leq x \leq N: C[x][x] \equiv_a 1)$$

$$Postocondición: \{(m)(MI) = C_{N\times N} \wedge IDENTIDAD(C,p)\}$$

Metodos auxiliares

encontrarDivisorPor1

E/S	Nombre	Tipo	Descripción
E	p	$p \in P$	Es número primo el cual se utiliza para realizar todas las fucniones módulo
E	R1	**nat**	Hace referencia a el valor de una celda de la matriz dada, además hace parte de la la diagonal de la matriz.
S	x	**nat**	Número x que satisface la ecuación R1$* \ x \ mod \ p = 1$

Precondición: $\{p \in P \ )\}$

PostCondición: $\{ \ x = (\exists \ x \in N \ \ |0 < x : R1 * x \ mod \ p = 1 \ \}$

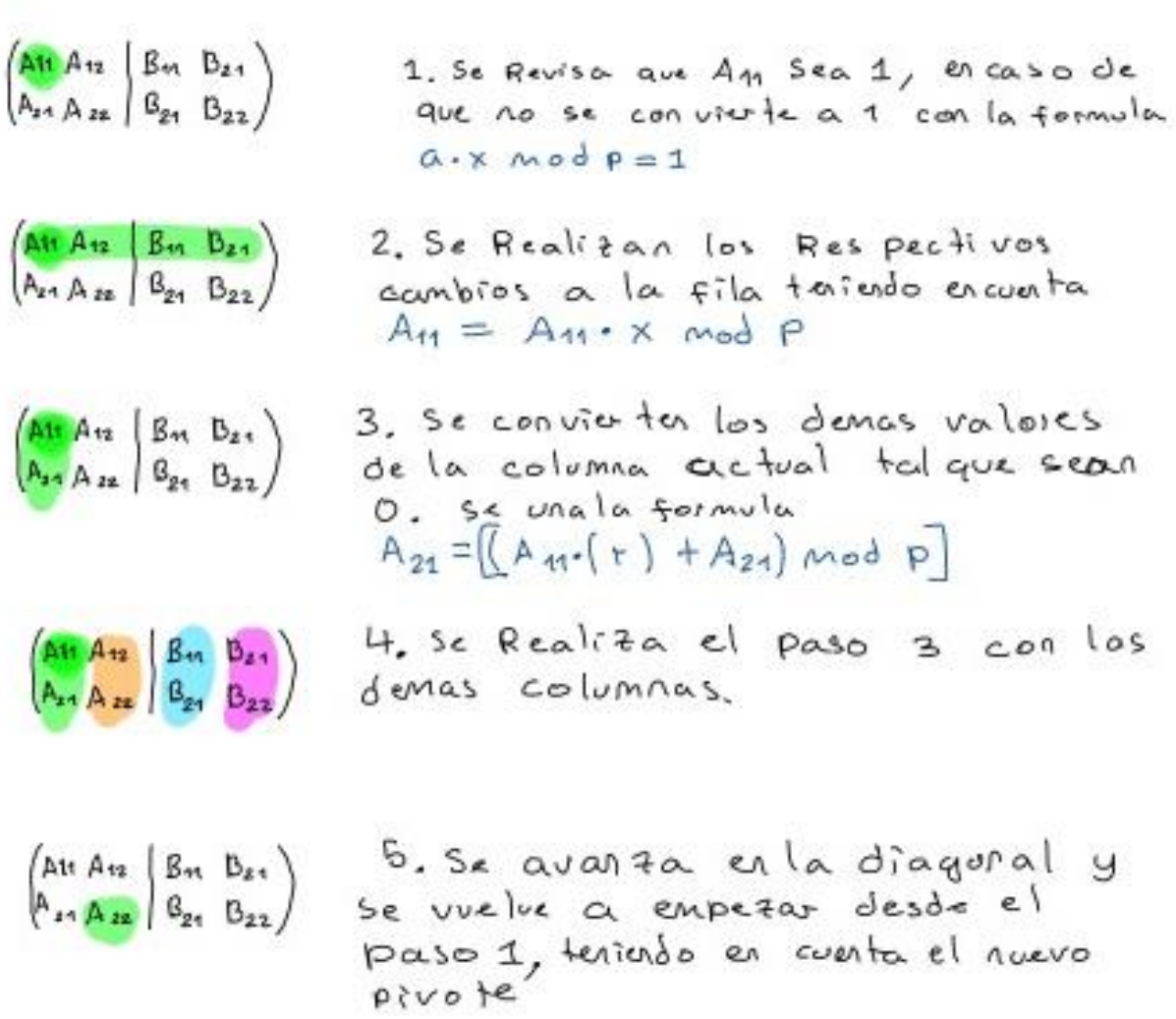
cambioDeFilas

E/S	Nombre	Tipo	Descripción
E	m	Array[0,N)[0,N)	Matriz de tamaño NxN.
E	fO	**nat**	Hace referencia a una de las filas que cambiara de posición con otra fila(fila destino)
E	FD	**nat**	Hace referencia a una de las filas que cambiara de posición con otra fila(fila origen)
S	m	Array[0,N)[0,N)	Matriz de tamaño NXN

Precondición: { $(\exists\, fO, FD \mid 0 < fO \leq N\,, 0 < fD \leq N : Fo < FD)$}

PostCondición: { $(\forall\, fO, FD \mid 0 < fO \leq N\,, 0 < fD \leq N : FO \rightarrow\ FD\ \wedge FD \rightarrow FO)$}

Explicación del algoritmo intuitiva



2. Análisis de complejidades espacial y temporal

- Cálculo de complejidades y explicación de estas.

Instrucción	Símbolo	Constante
Asignación	$=$	k_1

Suma	$+$	k_2
new	new	k_3
Menor	$<$	k_4
Mayor igual	$\geq$	k_5
.length	$.length$	k_6
Mayor	$>$	k_7
AND lógico	$\&\&$	k_8
minPositive	N/A	k_9
Resta	$-$	k_{10}
Equivalencia	$==$	k_{11}
Incremento	$++$	k_{12}
Decremento	$--$	k_{13}
Multiplicación	*	K14
Modulo	%	K15
Diferente	!	K16

$$T(n) = 2k1 + k6 + k3 + n(4k1 + 2k4 + 2k12) + n(4k1 + k4 + k12 + 2k11 + k16 + n(4k1 + k16 + +k4 + k7\ ) + n - h(4k1 + \ +k7 + 2k12 + \ +k16\, n(6k1 + k7 + 2k15 + +3k11 + k14 +) + n(6k1 + k7 + 2k15 + +3k11 + k14+\ )\ ))$$

$$T(n) = n * n * n * n - h * (n - j)^2$$

$$T(n) = n^3 * (n - h) * (n - j)^2$$

$$\boldsymbol{Orden\ de\ complejidad\ estimado \approx O(n^3 * (n - h) * (n - j)^2)}$$

$$Espacio(n) = O(n^2); Dos\ matrices\ cuadradas\ de\ tamaño\ n$$

Metodo encontrarDivisorPor1

$$T(n) = 2(k2 + k1 + k6) + 2n(k1 + k2 + k6 + +k1) +$$

$$T(n) = 2n \rightarrow O(n)$$

Metodo cambioDeFilas

$$T(n) = (k1 + k1) + 2n(k1 + k4 + k2 + K16 + k15 + 3K1)$$

$$T(n) = 2n \rightarrow O(n)$$

3. Comentarios finales

Pudimos darnos cuenta de que el algoritmo a pesar de que tiene muchos recorridos (for) estos se logran ejecutar en un tiempo razonable, además como a medida que se avanza en la matriz por la diagonal son menos los valores que se deben calcular.