**ISIS 1105 – 2020-2**

**Problema A – Inversa Matricial en Zp**

1. **Identificación**
   * Julián Oliveros – je.oliverosf – 201821595
   * Camilo Rozo – ce.rozob – 201820147
2. **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 () y un número primo (, calcular la matriz inversa multiplicativa correspondiente, si ésta existe. Se tomo la decisión de implementar un algoritmo de Gauss con unas variaicónes en como se realizaban la operaciones matemáticas, pues para la divición o multiplicación se implemento el terorema de Fermat 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 siaguiente ecuación , esta operación es equivalente la operación en el metodo Gauss de , Donde R2 es la casilla a la que quiere convirtir 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 unicas variaciones que se realizaron frente a el algoritmo de Eliminación de Gauss-Jordan.

En un pricipio se tomo 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 colomnas posteriores ya no eran relevantes, por lo que se fue iterando en la diagonal de la matriz conviertiendo estas casillas en pivotes, para despues convertir los demas valores de la columna en 0 y reliazar 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 , pues su función es encontrar un x tal que se satisfaga la ecuación.

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | Nombre | Tipo | Descripción |
| E | p |  | Es número primo el cual se utiliza para realizar todas las fucniones 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:

PostCondición:

**Metodos auxiliares**

**encontrarDivisorPor1**

|  |  |  |  |
| --- | --- | --- | --- |
| E/S | Nombre | Tipo | Descripción |
| E | 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 |

Precondición:

PostCondición:

**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:

PostCondición:

**Explicación del algoritmo intuitiva**

1. **Análisis de complejidades espacial y temporal**
   * Cálculo de complejidades y explicación de estas.

|  |  |  |
| --- | --- | --- |
| Instrucción | Símbolo | Constante |
| Asignación |  |  |
| Suma |  |  |
| new |  |  |
| Menor |  |  |
| Mayor igual |  |  |
| .length |  |  |
| Mayor |  |  |
| AND lógico |  |  |
| minPositive |  |  |
| Resta |  |  |
| Equivalencia |  |  |
| Incremento |  |  |
| Decremento |  |  |
| Multiplicación | \* | K14 |
| Modulo | % | K15 |
| Diferente | ! | K16 |

**Metodo encontrarDivisorPor1**

**Metodo cambioDeFilas**

1. **Comentarios finales**

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