Práctica 8

Cálculo de la matriz inversa en $\mathbb{Z}_m$

Podemos observar, viendo la tabla de multiplicar de $\mathbb{Z}_5$ en la tabla 1, todos los números no nulos de $\mathbb{Z}_5$ son inversibles: para cada uno de ellos siempre hay otro que multiplicado por él da como resultado la unidad. Así, el inverso de 2 módulo 5 es el 3 ya que $3\times 2 = 6 = 1 (\mathrm{mod}\, 5)$.

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Tabla 1: Tabla de multiplicar módulo 5

Sin embargo, como podemos ver con la tabla de multiplicar en $\mathbb{Z}_6$ mostrada en la tabla 2, no todos los números no nulos de $\mathbb{Z}_6$ son inversibles. El 5 sí es inversible ya que $5\times 5 = 25 = 1 (\mathrm{mod}\, 6)$, sin embargo, 2 no es inversible (no hay ningún número que multiplicado por 2 dé como resultado 1)

×	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Tabla 2: Tabla de multiplicar módulo 6

Para averiguar si un número es inversible o no y, en caso afirmativo, calcular el inverso, se puede utilizar el siguiente programa que calcularía todas las posibilidades (método de la fuerza bruta) hasta atinar con el inverso de un número x módulo m si es que existe:

```
%Método de la fuerza bruta para calcular el inverso
m=input('introduce el modulo en el que quieres trabajar ');
x= input('introduce el número al que le quieres calcular el inverso');
inverso=0;
for inverso=1:m-1
   if mod(inverso*x,m)==1
     break;
   end
end

if inverso==0
  disp('no existe inverso');
else
  disp(inverso);
end
```

En el caso anterior, para $x=120$ y para $m=131$, el programa devuelve 119.

Efectivamente podemos ver que $120\times119=1\,(\mathrm{mod}\,131)$

Basta con ejecutar en Matlab: mod(120*119, 131) y comprobar que efectivamente devuelve un 1.

Pero no hace falta aplicar el método de la fuerza bruta para saber si un número es o no inversible módulo m: un número x es inversible en $\mathbb{Z}_m$ si y solo si **es primo relativo con el módulo** m, es decir que el máximo común divisor de ambos debe ser 1:

$$mcd\,(m, x)=1$$

En el caso particular de que el módulo m sea un número primo, todos los elementos no nulos de $\mathbb{Z}_m$ serán primos relativos con él y, por lo tanto, serán inversibles.

Matlab trae incorporada una función que calcula el máximo común divisor de dos números: la función "gcd" (greatest common divisor). Así, por ejemplo:

$$gcd(6,4)=2$$

Hemos visto anteriormente cómo podemos averiguar el inverso de un número probando todas las opciones (método de la fuerza bruta). Sin embargo, existe una forma más sencilla de calcularlo usando la identidad de Bezout que has estudiado en Matemática Discreta.

Si $d=\mathrm{mcd}(x, y)$ entonces seguro que existen dos números enteros a y b tales $ax+by=d$

Estos números a y b se calculan usando el algoritmo extendido de Euclides.

Afortunadamente Matlab nos calcula los elementos de la identidad de Bezout, basta con ejecutar la orden:

$$[d\ a\ b] = gcd(x, y)$$

Por ejemplo, si ejecutamos en Matlab: $[d\ a\ b] = gcd(2\ ,5)$

el programa nos devuelve: $d = 1,\ a = -2$ y $b = 1$ ya que $2\times(-2)+5\times(1)=1$.

Si en la expresión anterior tomamos módulo 5 nos queda:

$$2\times(-2)+0\times(1)=1\,(\text{mod}\,5) \Rightarrow 2\times(-2)=1\,(\text{mod}\,5) \Rightarrow 2^{-1}=-2=3\,(\text{mod}5)$$

A continuación, escribimos una función en Matlab que devuelve el inverso de un número si es que existe, en caso contrario devuelve -999, un número ficticio que sólo sirve para indicar que no hay inverso.

```
function inv=calculo_inverso(x,m)
[d a b]=gcd(x,m);
  if d==1
       inv=mod(a,m);
  else
       inv=-999;
  end
```

También se puede trabajar con matrices en $\mathbb{Z}_m$. Las operaciones de sumar, multiplicar, etc. son las mismas de siempre salvo que la aritmética se hace módulo m.

Sabemos que una matriz es inversible cuando su determinante es no nulo cuando trabajamos con aritmética habitual. ¿Pero qué pasa cuando trabajamos en $\mathbb{Z}_m$?

Sabemos que una matriz A es inversible si y sólo si tiene determinante no nulo y que su inversa viene dada por: $A^{-1} = \frac{1}{|A|}(adj(A))^t$

Cuando trabajamos módulo m también podremos aplicar la fórmula anterior pero será necesario que $|A|$ sea un número inversible módulo m, por lo tanto, el $|A|$ debe ser primo relativo con m.

Por ejemplo, la inversa de una matriz de orden 2 se calcula de la siguiente forma:

$$\text{Si } A=\begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ y } |A|\neq 0 \Rightarrow A^{-1}=\frac{1}{|A|}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Eso sería en la aritmética habitual pero ahora será necesario que el número $|A|$ sea inversible en $\mathbb{Z}_m$.

Por ejemplo, la matriz: $A=\begin{pmatrix}4 & 2\\ 1 & 3\end{pmatrix}$ no será inversible módulo 15 ya que el determinante vale 10 que no es primo relativo con el módulo.

Sin embargo, la matriz $A=\begin{pmatrix}4 & 2\\ 1 & 3\end{pmatrix}$ sí que será inversible módulo 7 ya que: $|A|=10=3\,(\text{mod}\,7)$ y 3 tiene inverso módulo 7 que es el 5 ya que: $3\times 5=1\,(\text{mod}\,7)$.

Además la inversa se calcularía de la siguiente forma:

$$A^{-1}=\frac{1}{10}\begin{pmatrix}3 & -2\\ -1 & 4\end{pmatrix}=\frac{1}{3}\begin{pmatrix}3 & -2\\ -1 & 4\end{pmatrix}=5\begin{pmatrix}3 & -2\\ -1 & 4\end{pmatrix}=\begin{pmatrix}15 & -10\\ -5 & 20\end{pmatrix}=\begin{pmatrix}1 & 4\\ 2 & 6\end{pmatrix}(\text{mod}\,7)$$

Ya que el inverso de 3 módulo 7 es 5.

Podemos comprobar con Matlab cómo: $A\,A^{-1}=I\ (\text{mod}\,7)$, bastaría ejecutar:

```
A=[4 2;1 3];
B=[1 4;2 6];
mod(A*B, 7)
```

También podríamos haber calculado la inversa de A por operaciones elementales, pero haciendo las operaciones en $\mathbb{Z}_7$

$$\left(\begin{array}{cc|cc}4 & 2 & 1 & 0\\ 1 & 3 & 0 & 1\end{array}\right)\xrightarrow{f_1\leftrightarrow f_2}\left(\begin{array}{cc|cc}1 & 3 & 0 & 1\\ 4 & 2 & 1 & 0\end{array}\right)\xrightarrow{f_2\to f_2-4f_1}\left(\begin{array}{cc|cc}1 & 3 & 0 & 1\\ 0 & -10 & 1 & -4\end{array}\right)=\left(\begin{array}{cc|cc}1 & 3 & 0 & 1\\ 0 & 4 & 1 & 3\end{array}\right)(mod\ 7)$$

El inverso de 4 módulo 7 es 2, así que:

$$\left(\begin{array}{cc|cc}1 & 3 & 0 & 1\\ 0 & 4 & 1 & 3\end{array}\right)\xrightarrow{f_2\to f_2(2)}\left(\begin{array}{cc|cc}1 & 3 & 0 & 1\\ 0 & 1 & 2 & 6\end{array}\right)\xrightarrow{f_1\to f_1-f_2(3)}\left(\begin{array}{cc|cc}1 & 0 & -6 & -17\\ 0 & 1 & 2 & 6\end{array}\right)=\left(\begin{array}{cc|cc}1 & 0 & 1 & 4\\ 0 & 1 & 2 & 6\end{array}\right)(mod\ 7)$$

Ejercicio 1 (lápiz y papel)

Sea $H=\begin{pmatrix}1 & 1\\ 4 & 7\end{pmatrix}$ ¿será inversible módulo 31?

Hacer la inversa, si existe, por determinantes y después por operaciones elementales.

Sol. $H^{-1}=\begin{pmatrix}23 & 10\\ 9 & 21\end{pmatrix}$

Ejercicio 2 (lápiz y papel)

Calcular la inversa de $A=\begin{pmatrix}2 & 1 & 3\\ 2 & 1 & 2\\ 0 & 4 & 1\end{pmatrix}$ módulo 5.

Cálculo de la inversa módular con Matlab

Sabemos que la inversa de una matriz de orden 3 se calcula de la siguiente forma:

$$A^{-1} = \frac{1}{|A|}(adj(A))^t = \frac{1}{|A|}\begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{pmatrix}$$

siendo A_{ij} el adjunto del elemento a_{ij} de la matriz A.

Por ejemplo:

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 2 & 1 & 2 \\ 0 & 4 & 1 \end{pmatrix} \Rightarrow A^{-1} = \frac{1}{8}\begin{pmatrix} -7 & 11 & -1 \\ -2 & 2 & 2 \\ 8 & -8 & 0 \end{pmatrix}$$ en la aritmética habitual.

Si trabajamos módulo 5, el determinante es 8 que módulo 5 es 3 y 3 es primo relativo con el módulo 5 por lo tanto es inversible y su inverso es 2, entonces:

$$A^{-1} = \frac{1}{8}\begin{pmatrix} -7 & 11 & -1 \\ -2 & 2 & 2 \\ 8 & -8 & 0 \end{pmatrix} = 2\begin{pmatrix} -7 & 11 & -1 \\ -2 & 2 & 2 \\ 8 & -8 & 0 \end{pmatrix} = \begin{pmatrix} -14 & 22 & -2 \\ -4 & 4 & 4 \\ 16 & -16 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 4 & 4 \\ 1 & 4 & 0 \end{pmatrix} (\text{mod } 5)$$

El programa Matlab tiene una función "inv" que calcula la inversa de una matriz en la aritmética habitual. Así las órdenes de Matlab:

```
A=[ 2 1 3; 2 1 2; 0 4 1];
inv(A)
```

devuelven la matriz:

$$A = \begin{pmatrix} -0.875 & 1.375 & -0.125 \\ -0.25 & 0.25 & 0.25 \\ 1 & -1 & 0 \end{pmatrix}$$

Que efectivamente coincide con la expresión $A^{-1} = \frac{1}{8}\begin{pmatrix} -7 & 11 & -1 \\ -2 & 2 & 2 \\ 8 & -8 & 0 \end{pmatrix}$ anteriormente obtenida. ¿Cómo podemos usar Matlab para obtener la inversa de una matriz módulo m?

Si en el ejemplo anterior ejecutamos det(A)*inv(A) se obtiene la matriz: $\begin{pmatrix} -7 & 11 & -1 \\ -2 & 2 & 2 \\ 8 & -8 & 0 \end{pmatrix}$ es decir la transpuesta de la adjunta: $(adj(A))^t$.

Bastará con multiplicar por el inverso de 8 módulo 5 (que es un 2) y después tomar módulo 5. Así que la orden será: mod(2*(det(A)*inv(A)), 5)

Un programa para calcular la inversa de una matriz A módulo m sería:

```
m=5;
A=[2 1 3;2 1 2;0 4 1];
determinante=det(A);
[d a b]=gcd(mod(determinante,m),m);
if(d~=1)
    fprintf('la matriz no es inversible');
else
    B=round(determinante*inv(A));
    inverso_determinante=mod(a,m);
    B=mod(inverso_determinante*B,m);
    disp(B);
end
```

También se podría calcular la inversa usando operaciones elementales pero haciendo las operaciones módulo m.

Cifrado de texto con matrices

Supongamos que tenemos el siguiente alfabeto de 27 letras:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Vamos a cifrar un mensaje de texto: “HOLA”

Pues bien, “HOLA” $= \{7, 15, 11, 0\}$

Consideramos la matriz clave secreta: $K = \begin{pmatrix} 1 & 2 \\ 4 & 6 \end{pmatrix}$

Agrupamos de dos en dos el mensaje a cifrar y hacemos:

$$K\begin{pmatrix}7\\15\end{pmatrix} = \begin{pmatrix}1 & 2\\4 & 6\end{pmatrix}\begin{pmatrix}7\\15\end{pmatrix} = \begin{pmatrix}37\\118\end{pmatrix} = \begin{pmatrix}10\\10\end{pmatrix} \ (mod\ 27)$$

$$K\begin{pmatrix}11\\0\end{pmatrix} = \begin{pmatrix}1 & 2\\4 & 6\end{pmatrix}\begin{pmatrix}11\\0\end{pmatrix} = \begin{pmatrix}11\\44\end{pmatrix} = \begin{pmatrix}11\\17\end{pmatrix} \ (mod\ 27)$$

La lista {10, 10, 11, 17} se corresponde con "KKLQ" que será el mensaje cifrado.

Para descifrar el mensaje cifrado se hará lo siguiente:

- Se pasa el mensaje cifrado a números: "KKLQ"={10, 10, 11, 17}
- Se cogen de dos y en dos y hacemos:

  $K^{-1}\begin{pmatrix}10\\10\end{pmatrix} = \begin{pmatrix}7\\15\end{pmatrix} \ (mod\ 27)$

  $K^{-1}\begin{pmatrix}11\\17\end{pmatrix} = \begin{pmatrix}11\\0\end{pmatrix} \ (mod\ 27)$
- El mensaje es {7, 15, 11, 0}="HOLA"

Ejercicio 3 (lápiz y papel, ayudándote con Matlab)

Descifrar el siguiente mensaje "CGQQSA" sabiendo que se cifró con el alfabeto de 27 caracteres de la tabla anterior con la matriz de cifrado:

$$K = \begin{pmatrix}1 & 4\\3 & 1\end{pmatrix}$$

Ejercicio 4 (lápiz y papel, ayudándote con Matlab)

Con el alfabeto anterior de 27 caracteres y la matriz de cifrado:

$$K = \begin{pmatrix}8 & 6 & 9 & 5\\6 & 9 & 5 & 10\\5 & 8 & 4 & 9\\10 & 6 & 11 & 4\end{pmatrix}$$

Se ha obtenido el mensaje: "GWGKPGBWUERK".

Descifrar el mensaje.

Ejercicio 5 (lápiz y papel, ayudándote con Matlab)

A veces la clave se pone en formato de texto. Por ejemplo, podría ser la clave "MAGNOLIAS" y se sabe que se usa una matriz de cifrado 3×3 y con el alfabeto de 27 caracteres anterior. Esto equivale a usar la matriz de cifrado:

$$K = \begin{pmatrix}M & A & G\\N & O & L\\I & A & S\end{pmatrix}$$

Esta matriz se convierte en una matriz de números de acuerdo al alfabeto de 27 caracteres donde la M es 12, la A es 0, etc.

¿Es la matriz anterior una matriz válida de cifrado módulo 27?

Cifrado de imágenes digitales

Vamos a cifrar una imagen digital con escala de grises entre 0 y 255. Seguiremos una idea similar a la seguida en los ejercicios anteriores. Este método está inspirado en un trabajo de Lester Hill publicado en 1929 [1]. Se sigue investigando actualmente en este método de cifrado, como puedes ver en los artículos [2] y [3] publicados en 2010 y 2013 respectivamente.

Supongamos que indicamos por A a la matriz correspondiente a la imagen original. Vamos a coger los niveles de gris de dos en dos, empezando en la esquina superior izquierda de la matriz y moviéndonos de izquierda a derecha y de arriba a abajo: el primer bloque será $\{a_{11}, a_{12}\}$, el segundo bloque será $\{a_{13}, a_{14}\}$, y así sucesivamente.

Cada bloque de dos niveles de gris de la imagen original se va a transformar en otros dos números mediante un producto matricial con una matriz secreta K de tamaño 2×2 (esto por supuesto se puede cambiar y elegir como queremos). Supongamos que el bloque que estamos procesando es: $\{125, 137\}$ y que la matriz secreta es: $K = \begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}$

Hacemos el producto matricial: $\begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}\begin{pmatrix} 125 \\ 137 \end{pmatrix} = \begin{pmatrix} 740 \\ 13073 \end{pmatrix}$

Como vemos los números $\{740, 13073\}$ exceden el valor 255 y por lo tanto no se corresponden con niveles de gris. Para conseguir que el resultado proporcione números entre 0 y 255, tomaremos módulo 256:

$$\begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}\begin{pmatrix} 125 \\ 137 \end{pmatrix} = \begin{pmatrix} 7420 \\ 13073 \end{pmatrix} = \begin{pmatrix} 252 \\ 17 \end{pmatrix} (\text{mod } 256)$$

Ahora el píxel que tenía nivel de gris 125 lo pondremos a 252 y el píxel de al lado que tenía nivel de gris 137 será puesto a 17.

De esta manera, vamos transformando cada par de valores de gris consecutivos por otro par de valores de gris diferentes. De esta forma conseguimos cifrar la imagen. La imagen cifrada obtenida en este caso se muestra en a continuación:

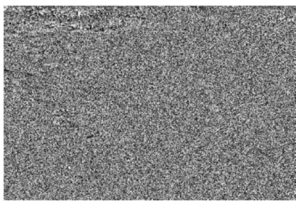


Fig. Imagen cifrada

Ejercicio 6 (ordenador)

Tu trabajo consistirá en tomar la imagen "matricial.png" de la página web y descifrarla sabiendo que hemos empleado en el cifrado la matriz $K = \begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}$ y módulo 256.

Nota: Si pones A=imread('matricial.png');

A será una matriz uint8 (como puedes ver con la orden whos A).

Para poder trabajar con ella no olvides poner A=double(A);

Ejercicio 7 (ordenador)

Supongamos que "imagen.bmp" es una imagen en color RGB de tamaño 512×512.

Por ejemplo, la una imagen de esas características se muestra a continuación:

Fig. Imagen original en color

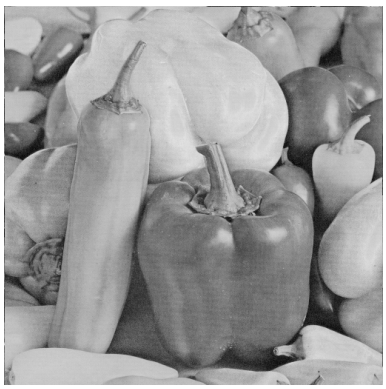
Entonces con la orden Matlab:

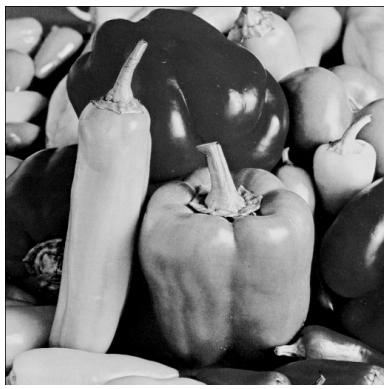
A=imread('imagen.bmp'); whos A

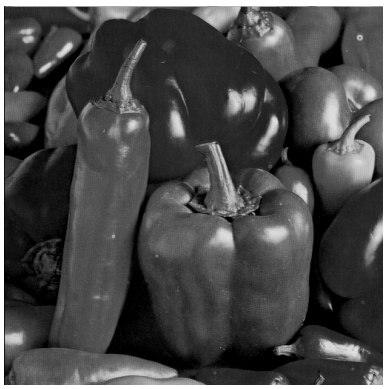
Verás que Matlab te dice que A es una estructura $512 \times 512 \times 3$.

Pues bien, la matriz $A(:,:,1)$ es la capa de rojo, $A(:,:,2)$ es la capa de verde y $A(:,:,3)$ es la capa de azul. Cada capa es una matriz de tamaño 512×512 con números entre 0 y 255 indicando la cantidad de rojo, verde y azul presente en el color del píxel. Por ejemplo, si un píxel fuera rojo puro, tendría 255 en la capa de rojo, cero en la capa de verde y 0 en la capa de azul.

Con la orden de Matlab imshow$(A(:,:,1))$ puedes mostrar la capa de rojo, con imshow$(A(:,:,2))$ la capa de verde y con imshow$(A(:,:,3))$ la capa de azul. Las mostramos a continuación:


Rojo


Verde


Azul

Cada capa, por tanto, es una imagen en escala de grises con números entre 0 y 255.

Podemos aplicarle a cada capa un cifrado diferente:

- Se crea una estructura del tipo: $cifrada = zeros(512, 512, 3)$
- Se cifra la capa $A(:,:,1)$ con una clave K_1 y su resultado se mete en $cifrada(:,:,1)$
- Se cifra la capa $A(:,:,2)$ con una clave K_2 y su resultado se mete en $cifrada(:,:,2)$
- Se cifra la capa $A(:,:,3)$ con una clave K_3 y su resultado se mete en $cifrada(:,:,3)$
- Con la orden imshow(uint8(cifrada)) podemos mostrar el resultado.

En Moodle tienes la imagen "cifrada_color.bmp" que se obtuvo cifrando la capa de rojo con la matriz clave $K_1 = \begin{pmatrix} 21 & 35 \\ 18 & 79 \end{pmatrix}$, la capa de verde con la matriz $K_2 = \begin{pmatrix} 11 & 48 \\ 3 & 119 \end{pmatrix}$ y la capa de azul con la capa $K_3 = \begin{pmatrix} 80 & 47 \\ 39 & 109 \end{pmatrix}$ y módulo 256.

Tu trabajo consistirá en tomar la imagen "cifrada_color.bmp" y descifrarla.

Bibliografía

[1] Hill, L.S.; "Cryptography in an algebraic alphabet". The American Mathematical Monthly, 38, pp. 135-154, 1929.

[2] Acharya B. et al. "Image encryption with advanced Hill Cipher algorithm". International Journal on Signal and Image Processing. Vol 1, No. 1, Jan 2010

[3] Nordin, A. et al. "Cryptography: A New Approach of Classical Hill Cipher", International Journal of Security and Its Applications. Vol. 7, No. 2, March, 2013