Guía de Estudio
Curso: Estructuras discretas
Tema: Propiedades de los enteros
Inducción matemática y Teoría de números

Guina Guadalupe Sotomayor Alzamora
Ingeniería de Sistemas, Universidad Nacional del Altiplano de Puno, Perú
gsotomayor@unap.edu.pe

7 de marzo de 2024

Objetivos de aprendizaje

Al terminar este capítulo, los alumnos deben ser capaces de:

- Comprender Inducción matemática y su uso.
- Comprender Divisibilidad y su uso.
- Comprender la idea de Criptografía.

1. Introducción

Tenemos conocimiento sobre los enteros desde nuestros primeros encuentros con la aritmética, se examinará una propiedad especial exhibida por el subconjunto de enteros positivos. Esta propiedad ayudará a establecer ciertas fórmulas y teoremas matemáticos usando una técnica llamada *inducción matemática*. Este método de prueba jugará un rol principal en muchos de los resultados que serán obtenido, así mismo se hará una introducción a cuatro conjuntos de números muy importantes en el estudio de matemáticas discretas, nombrados los números triangulares, los números armónicos, los números Fibonacci y los números Lucas.

Cuando $x, y \in \mathbb{Z}$, se sabe que $x+y$, $x \times y$, $x-y \in \mathbb{Z}$. Así, se puede decir que el conjunto $\mathbb{Z}$ es **cerrado** bajo (operaciones binarias de) adición, multiplicación y sustracción. Sin embargo, en la división, si se tiene que $2, 3 \in \mathbb{Z}$ pero el número racional $2/3 \notin \mathbb{Z}$. Así, el conjunto $\mathbb{Z}$ de todos los enteros **no es cerrado** bajo la operación binaria de división distinta de cero. Para hacer frente a esto se introducirá una forma restringida de división para $\mathbb{Z}$, concentrándonos en elementos especiales de $\mathbb{Z}^+$ llamados primos.

La inducción matemática se usa para probar si una expresión matemática (igualdad o desigualdad) es falsa o verdadera, sin necesidad de representarla con notación lógica. Es común desarrollar programas donde se tiene un *valor inicial* para la primera iteración, un incremento o decremento que puede ser aplicado por medio del *n-ésimo* término que permite obtener los valores de una sumatoria en cada iteración y un *resultado* de la sumatoria, el cual se puede representar de forma general por medio de una expresión matemática. Así, se puede representar algoritmos en forma matemática y probar si esos algoritmos son falsos o verdaderos, usando para ello la inducción matemática. Para usar la inducción matemática en la demostración de algoritmos se necesita que estos se representen como

una sumatoria de la forma:

$$\underbrace{x_1}_{\text{inicio}} + x_2 + x_3 + \cdots + \underbrace{k}_{\text{n-ésimo término}} = \underbrace{r}_{\text{resultado}}$$

El primer elemento x_1 se obtiene en la primera iteración ($n = 1$) y se conoce como valor inicial, el n-ésimo término permite encontrar cada uno de los elementos de la sumatoria, deberá estar en función de n, y depende del valor de n que permite encontrar el resultado de sumar los n elementos de la sumatoria. La sumatoria, incluyendo el inicio, el n-ésimo término y el resultado es la proposición $S(n)$.

2. Principio de la inducción matemática

Dados dos enteros distintos, x, y, se sabe que $x < y$ o $y < x$. Sin embargo, es también verdad que en lugar de tener enteros, x y y son números racionales o reales. Suponga que se quiere expresar el subconjunto $\mathbb{Z}^+$ de $\mathbb{Z}$, usando símbolos de inecuaciones $>$ y $\geq$. Así, el conjunto de los elementos positivos de $\mathbb{Z}$ es:

$$\mathbb{Z}^+ = \{x \in \mathbb{Z} | x > 0\} = \{x \in \mathbb{Z} | x \geq 1\}.$$

Sin embargo, cuando se hace lo mismo con números reales o racionales se tiene:

$$\mathbb{Q}^+ = \{x \in \mathbb{Q} | x > 0\} \quad \text{and} \quad \mathbb{R}^+ = \{x \in \mathbb{R} | x > 0\},$$

pero no se puede representar $\mathbb{Q}^+$ o $\mathbb{Q}^+$ usando $\geq$ como con $\mathbb{Z}^+$.

Sea $S(n)$ una declaración matemática abierta, que envuelva una o más ocurrencias de la variable n, que representa un entero positivo.

1. Si $S(1)$ es verdad; y
2. Si $S(k)$ es verdad ($k \in \mathbb{Z}^+$) entonces $S(k+1)$ es verdad;

finalmente $S(n)$ es verdad para todo $n \in \mathbb{Z}^+$

La selección del 1 en la primera condición no es obligatoria, pero es necesario que exista un primer elemento para que el proceso de inducción comience, un $n_0 \in \mathbb{Z}$. El principio de inducción matemática puede ser expresado, usando cuantificadores, como:

$$[S(n_0) \wedge [\forall k \geq n_0 \; [S(k) \implies S(k+1)]]] \implies \forall n \geq n_0 \; S(n).$$

Ejemplos:

1. Para todo $n \in \mathbb{Z}^+, \sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n = \dfrac{n(n+1)}{2}$,

   Para $n = 1$, como base inductiva se tiene:

   $$S(1) : \sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} = 1$$

   Así, $S(1)$ es verdad.

   Si $(n = k)$, se asume que sea verdad como hipótesis de inducción, para establecer se necesita establecer la verdad para $S(k+1)$ como tesis, entonces se necesita demostrar que la tesis es verdadera a partir de la hipótesis dada.

$$S(k+1): \sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

Entonces:

$$\sum_{i=1}^{k+1} i = 1+2+3+\cdots+k+(k+1) = \left(\sum_{i=1}^{k} i\right)+(k+1) = \frac{k(k+1)}{2}+(k+1)$$

porque se está asumiendo la verdad de $S(k)$. Pero:

$$\frac{k(k+1)}{2}+(k+1) = \frac{k(k+1)}{2}+\frac{2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$$

estableciendo el paso inductivo del teorema.

Por lo tanto, por el principio de Inducción Matemática, $S(n)$ es verdadero para todo $n \in \mathbb{Z}^+$. Vale indicar que n es el n-ésimo *número triangular* (cuyos elementos pueden ser dispuestos en un triángulo equilátero).

2. Probar que para cada $n \in \mathbb{Z}^+$,

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

Para $n = 1$

$$S(1): \sum_{i=1}^{1} i^2 = \frac{1(1+1)(2\times 1+1)}{6} = \frac{(2)(3)}{6} = 1$$

Así, $S(1)$ es verdad.

Asumiendo que $S(k)$ sea verdad para $k \in \mathbb{Z}^+$ se asume que:

$$S(k): \sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$$

sea verdad. Usando la hipótesis de inducción $S(k)$, se tiene que:

$$\begin{aligned} S(k+1): \quad \sum_{i=1}^{k+1} i^2 &= \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6} \\ &= \frac{(k+1)(k+2)(2k+3)}{6} \end{aligned}$$

estableciendo el paso inductivo del teorema.

$$\begin{aligned} \sum_{i=1}^{k+1} i^2 &= 1^2+2^2+\cdots+k^2+(k+1)^2 = \sum_{i=1}^{k} i^2+(k+1)^2 \\ &= \left[\frac{k(k+1)(2k+1)}{6}\right]+(k+1)^2 \\ &= (k+1)\left[\frac{k(2k+1)}{6}+(k+1)\right] \\ &= (k+1)\left[\frac{2k^2+7k+6}{6}\right] = \frac{(k+1)(k+2)(2k+3)}{6} \end{aligned}$$

El resultado general sigue el principio de Inducción Matemática, $S(n)$ es verdadero para todo $n \in \mathbb{Z}^+$.

3. Entre las secuencias más interesantes de números encontrados en matemáticas discretas se tiene a los *números armónicos* $H_1, H_2, H_3, \ldots$, donde:

$$\begin{aligned} H_1 &= 1 \\ H_2 &= 1 + \frac{1}{2} \\ H_3 &= 1 + \frac{1}{2} + \frac{1}{3} \\ &\ldots \end{aligned}$$

y en general $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$, $\forall n \in \mathbb{Z}^+$. Se tiene que $\forall n \in \mathbb{Z}^+$:

$$\sum_{j=1}^{n} H_j = (n+1)H_n - n$$

Probando para S(1):

$$\sum_{j=1}^{1} H_j = H_1 = 1 = 2 \times 1 - 1 = (1+1)H_1 - 1$$

Para verificar el paso inductivo, se asume la verdad de $S(k)$, esto es:

$$\sum_{j=1}^{k} H_j = (k+1)H_k - k$$

Luego:

$$\begin{aligned} \sum_{j=1}^{k+1} H_j = \sum_{j=1}^{k} H_j + H_{k+1} &= [(k+1)H_k - k] + H_{k+1} \\ &= (k+1)H_k - k + H_{k+1} \\ &= (k+1)[H_{k+1} - (1/(k+1))] - k + H_{k+1} \\ &= (k+2)H_{k+1} - 1 - k \\ &= (k+2)H_{k+1} - (k+1) \end{aligned}$$

Consecuentemente, por el principio de inducción matemática $S(n)$ es verdadero para todos los enteros positivos n.

4. El factorial de n (o n factorial) se define como:

$$n! = \begin{cases} 1 & \text{si} \quad n = 0 \\ n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1 & \text{si} \quad n \geq 1 \end{cases}$$

Si $n \geq 1$, $n!$ es igual al producto de todos los enteros entre 1 y n inclusive, teniendo como caso especial 0!, definido como 1.

Por inducción se espera demostrar que

$$S(n) : n! \geq 2^{n-1} \ \forall n \geq 1$$

Para S(1):

$$S(1) : 1! = 1 \geq 1 = 2^{1-1}$$

Se asume que sea verdad que S(k):

$$S(k) : k! \geq 2^{k-1}$$

Se debe probar que para S(k+1):

$$S(k+1) : (k+1)! \geq 2^k$$

Se puede relacionar $S(k)$ y $S(k+1)$ si se observa que

$$(k+1)! = (k+1) \times (k!)$$

Además

$$\begin{aligned} (k+1)! &= (k+1) \times (k!) \\ &\geq (k+1) \times 2^{k-1} \\ &\geq 2 \times 2^{k-1} \qquad \text{ya que } (k+1) \geq 2 \\ &\geq 2^k \end{aligned}$$

Ejercicios

Demuestre que:

1. $2 + 5 + 8 + \cdots + (3n-1) = \dfrac{n(3n+1)}{2}$
2. $1 + 5 + 9 + \cdots + (4n-3) = n(2n-1)$
3. $1 + a + a^2 + a^3 + \cdots + a^{n-1} = \dfrac{a^n - 1}{a-1}, a \neq 1$

3. Definiciones recursivas

Las demostraciones por recursividad tienen que basarse en la definición del dominio, porque estas definiciones proporcionan las premisas necesarias, sobre las cuales se pueden construir las demostraciones. Por ejemplo: Si x es una persona, entonces, por definición, todas las personas siguientes son descendientes de x.

1. Todos los hijos de x son descendientes de x
2. Si y es un hijo de x, entonces todos los descendientes de y son descendientes de x
3. No hay nadie más que sea descendiente de x

Para que una definición recursiva sea válida, no debe generar tipos de datos o funciones con ciclos infinitos de evaluación, así, debe constar de dos partes:

1. Un conjunto de casos base, que son simples, con una definición directa y sin usar autoreferencia.
2. Un conjunto de reglas recursivas, aquí se define un nuevo elemento de la definición, en términos anteriores que ya hayan sido definidos.

Además, la definición debe constar de una cláusula que asegure que las dos anteriores son las únicas formas de obtener el concepto, objeto o función definida, esta puede omitirse pues se entiende que siempre está presente. Por ejemplo, si se define la función

$$f(0) = 1$$

$$f(n+1) = f(n+2)$$

no es válida, pues la definición en $n+1$ está dada en términos de un elemento posterior a $n+1$, a saber $n+2$. Así, f resulta indefinida en cualquier valor diferente a cero, siendo una *recursiva general* y por lo general estas definiciones causan ciclos infinitos en programación. Algunos ejemplos se muestran a continuación:

1. Considere la secuencia entera $b_0, b_1, b_2, b_3, \ldots$, donde $b_n = 2n \ \forall n \in \mathbb{N}$. Entonces se tiene: $b_0 = 2 \times 0 = 0$, $b_1 = 2 \times 1 = 2$, $b_2 = 2 \times 2 = 4$, $b_3 = 2 \times 3 = 6$, etc. Si se solicita determinar b_6 la forma más simple es calcular $b_6 = 2 \times 6 = 12$, sin necesitar calcular el valor de b_n para algún otro $n \in \mathbb{N}$, esto dado que tenemos una fórmula explícita llamada $b_n = 2n$ que determina como encontrar b_n a partir de sólo n.

2. Considere la secuencia de enteros $a_0, a_1, a_2, a_3, \ldots$, donde:

   a) $a_0 = 1, a_1 = 2, a_2 = 3$, y

   b) $a_n = a_{n-1} + a_{n-2} + a_{n-3} \ \forall n \in \mathbb{Z}^+$ donde $n \geq 3$.

   No se tiene una fórmula explícita que defina cada a_n en términos de $\forall n \in \mathbb{N}$. En este caso, si se quiere evaluar a_6, se necesita conocer los valores de a_5, a_4 y a_3. Además, estos valores (a_5, a_4 y a_3) requieren también conocer los valores de a_2, a_1 y a_0. Así, para encontrar el valor de a_6 se tiene:

$$\begin{aligned}
a_6 &= a_5 + a_4 + a_3 \\
&= (a_4 + a_3 + a_2) + (a_3 + a_2 + a_1) + (a_2 + a_1 + a_0) \\
&= [(a_3+a_2+a_1)+(a_2+a_1+a_0)+a_2]+[(a_2+a_1+a_0)+a_2+a_1]+(a_2+a_1+a_0) \\
&= [((a_2 + a_1 + a_0) + a_2 + a_1) + (a_2 + a_1 + a_0) + a_2] + [(a_2 + a_1 + a_0) + a_2 + \\
&\quad a_1] + (a_2 + a_1 + a_0) \\
&= [((3+2+1)+3+2)+(3+2+1)+3]+[(3+2+1)+3+2]+(3+2+1) \\
&= [((6)+5)+(6)+3]+[(6)+5]+(6) \\
&= [20]+[11]+(6) = 37
\end{aligned}$$

   Otra forma de resolver el problema es seguir los siguientes pasos:

$$\begin{aligned}
a_3 &= a_2 + a_1 + a_0 = 3 + 2 + 1 = 6 \\
a_4 &= a_3 + a_2 + a_1 = 6 + 3 + 2 = 11 \\
a_5 &= a_4 + a_3 + a_2 = 11 + 6 + 3 = 20 \\
a_6 &= a_5 + a_4 + a_3 = 20 + 11 + 6 = 37
\end{aligned}$$

3. Los *números Fibonacci* también pueden ser definidos recursivamente por:

   a) $F_0 = 0, F_1 = 1$, y

   b) $F_n = F_{n-1} + F_{n-2} \ \forall n \in \mathbb{Z}^+$ donde $n \geq 2$.

   Por ejemplo, los primeros 10 números Fibonacci serán $F = \{0, 1, 1, 2, 3, 5, 8, 13, 21, 34\}$.

4. La secuencia de los *números Lucas*, secuencia muy parecida a Fibonacci, también pueden ser definidos recursivamente por:

a) $L_0 = 2, L_1 = 1$, y

b) $L_n = L_{n-1} + L_{n-2}\ \forall n \in \mathbb{Z}^+$ donde $n \geq 2$.

Aunque los números Lucas no sean tan conocidos como los Fibonacci, poseen propiedades interesantes, como la interrelación entre los números Fibonacci y Lucas, que indica:

$$\forall n \in \mathbb{Z}^+\ L_n = F_{n-1} + F_{n+1}$$

Por ejemplo, los primeros 10 números lucas será: $L = \{2, 1, 3, 4, 7, 11, 18, 29, 47, 76\}$.

Ejercicios

Programe de forma recursiva:

1. El Factorial de un número $n \in \mathbb{Z}^+$
2. Los $n \in \mathbb{Z}^+$ números triangulares requeridos
3. Los $n \in \mathbb{Z}^+$ números armónicos requeridos
4. Los $n \in \mathbb{Z}^+$ números Fibonacci requeridos
5. Los $n \in \mathbb{Z}^+$ números Lucas requeridos a partir de números Fibonacci

4. División en los enteros

La teoría de números es una rama de las matemáticas *puras*, conocida más por su naturaleza que por sus aplicaciones. A finales del siglo XX esta ha adquirido mayor importancia en los sistemas criptográficos, que son sistemas usados para la seguridad en las comunicaciones matemáticas que se ocupa de los números enteros.

La propiedad del buen orden para enteros no negativos establece que todo conjunto no vacío de enteros no negativos tiene un elemento menor. Esta propiedad es equivalente a las dos formas de inducción. Usando la propiedad del buen orden para probar algo familiar de la división se tiene que cuando se divide un entero n (dividendo) entre un entero positivo d (divisor), se obtiene el cociente q y un residuo r que satisface $0 \leq r < d$ de manera que $n = dq + r$, vale indicar que $r = 0$ sí y sólo sí d divide a n. Por ejemplo, cuando se divide $n = 73$ entre $d = 13$:

$$\begin{array}{r|l} 73 & 13 \\ \hline 8 & 5 \end{array}$$

Se obtuvo un cociente $q = 5$ y un residuo de $r = 8$. Se observa que r satisface: $0 \leq 8 < d$, esto es $0 \leq 8 < 13$. Finalmente:

$$n = 73 = 13 \times 5 + 9 = d \times q + r$$

Un entero mayor que 1 cuyos únicos divisores positivos son 1 y él mismo se llama *primo*. Un entero mayor que 1 que no es primo se llama *compuesto*. Por ejemplo, el entero 17 es primo, pues sus únicos divisores son 1 y él mismo, y el entero 26 es compuesto, pues es divisible entre 13 que no es ni 1 ni 26. Así, para determinar si un entero positivo $n > 1$ es primo, se verifican los divisores potenciales $2, 3, \ldots, n-1$, sin embargo, en realidad es suficiente con verificar:

$$2, 3, \ldots, \sqrt{n},$$

cuyo algoritmo se resume como sigue:

Algoritmo 1: EsPrimo(n)

entrada: n
salida : d
for $d = 2$ *to* $\sqrt{n}$ **do**
  if *(n mod d) == 0* **then** return d;
return 0;

Este algoritmo determina si el entero $n > 1$ es primo. Si n es primo, el algoritmo retorna 0. Si n es compuesto, el algoritmo regresa un divisor d que satisface $2 \leq d \leq \sqrt{n}$. Para probar si d divide a n, el algoritmo verifica si el residuo al dividir n entre d, esto es n mod d.

Ejercicios

Determine:

1. si los siguientes números son primos o compuestos: 1, 57, 83, 123, 143
2. los primeros 10 números primos
3. los números primos menores a 50

5. Máximo Común Divisor

Si x, y y k están en $\mathbb{Z}^+$, y $k|x$, $k|y$, se dice que k es un **divisor común** de x y y. Si d es el mayor de estos k, a d se le llama **máximo común divisor**, o MCD, de x y y, y se escribe MCD(x, y). Osea, k es un entero que divide tanto a x como a y, y su mayor valor es el MCD. Entonces, si d es el MCD(x, y), entonces:

1. MCD(x, y)= $sx + ty$ para algunos enteros s y t (que no son ambos positivos).
2. Si c es cualquier otro divisor común de x y y, entonces $c|d$

Por ejemplo:

1. Sean los números 12 y 30, los divisores de 12 son: 1, 2, 3, 4, 6, 12; los divisores de 30 son: 1, 2, 3, 5, 6, 10, 15, 30; los divisores comunes de 12 y 30 son: 1, 2, 3, 6, por lo que el MCD(12, 30)=30 y $6 = (-2) \times 12 + 1 \times 30$.
2. Sean los números 13 y 27, los divisores de 13 son 1 y 13; los divisores de 27 son: 1, 3, 9, 27; el divisor de ambos es el 1, ya que el 13 es primo, se verifica que 1= $13 \times (-2) + 27 \times 1$.
3. Sean los números 30 y 105, los divisores de 30 son: 1, 2, 3, 5, 6, 10, 15, 30; los divisores de 105 son: 1, 3, 5, 7, 15, 21, 35, 105; los divisores comunes de 30 y 105 son: 1, 3, 5, 15; se concluye que MCD(30,105), es 15 y que $15 = 30 \times 4 + 105 \times (-1)$.

Un algoritmo antiguo, conocido y *eficiente* para encontrar el máximo común divisor de dos enteros es el **Algoritmo euclidiano**, en el cual se determina el MCD(x, y). Aquí, en principio se supone que $x > y > 0$, de lo contrario, x, y se intercambian. Se tiene que:

$$x = k_1 y + r_1 \text{ en donde } k_1 \text{ y } r_1 \text{ están en } \mathbb{Z}^+ \text{ y } 0 \leq r_1 < y$$

Además se sabe que si n divide a x y y, se debe dividir a r_1, ya que $r_1 = a - k_1 \times y$. De modo similar, si n divide a y y a r_1, se debe dividir a x. Los divisores comunes de x y y son los mismos que los divisores comunes de y y r_1, así MCD(x, y)=MCD(b, r_1). Luego se continua como sigue:

$$
\begin{array}{ccc}
\text{divida } y \text{ entre } r_1 & y = k_2 r_1 + r_2 & 0 \leq r_2 < r_1 \\
\text{divida } r_1 \text{ entre } r_2 & r_1 = k_3 r_2 + r_3 & 0 \leq r_3 < r_2 \\
\text{divida } r_2 \text{ entre } r_3 & r_2 = k_4 r_3 + r_4 & 0 \leq r_4 < r_3 \\
\ldots & \ldots & \ldots \\
\text{divida } r_{n-2} \text{ entre } r_{n-1} & r_{n-2} = k_n r_{n-1} + r_n & 0 \leq r_n < r_{n-1} \\
\text{divida } r_{n-1} \text{ entre } r_n & r_{n-1} = k_{n+1} r_n + r_{n+1} & 0 \leq r_{n+1} < r_n
\end{array}
$$

Como $x > y > r_1 > r_2 > r_3 > r_4 > \ldots$, el residuo tendrá que llegar a ser cero, por lo que en algún punto se obtendrá $r_{n+1} = 0$. Por ejemplo:

1. Sean x=105 y y=30, se tiene:

$$
\begin{array}{cc}
\text{se divide 105 entre 30} & 105 = 3 \times 30 + 15 \\
\text{se divide 30 entre 15} & 30 = 2 \times 15 + 0
\end{array}
$$

   De manera que MCD(105, 30)=15, el último de los divisores.

2. Sean x=190 y y=34, se tiene

$$
\begin{array}{cc}
\text{se divide 190 entre 34} & 190 = 5 \times 34 + 20 \\
\text{se divide 34 entre 20} & 34 = 1 \times 20 + 14 \\
\text{se divide 20 entre 14} & 20 = 1 \times 14 + 6 \\
\text{se divide 14 entre 6} & 14 = 2 \times 6 + 2 \\
\text{se divide 6 entre 2} & 6 = 3 \times 2 + 0
\end{array}
$$

   De manera que MCD(190, 34)=2, el último de los divisores.

3. Sean x=540 y y=504, se tiene

$$
\begin{array}{cc}
\text{se divide 540 entre 504} & 540 = 1 \times 504 + 36 \\
\text{se divide 504 entre 36} & 504 = 14 \times 36 + 0
\end{array}
$$

   De manera que MCD(540,504)=36, el último de los divisores.

4. Sean x=660 y y=29, se tiene

$$
\begin{array}{cc}
\text{se divide 660 entre 29} & 660 = 22 \times 29 + 22 \\
\text{se divide 29 entre 22} & 29 = 1 \times 22 + 7 \\
\text{se divide 22 entre 7} & 22 = 3 \times 7 + 1 \\
\text{se divide 7 entre 1} & 7 = 7 \times 1 + 0
\end{array}
$$

   De manera que MCD(660,29)=1, el último de los divisores.

Los valores para s y t se pueden calcular usando el algoritmo euclidiano, por ejemplo, para $x = 660$ y $y = 29$ se trabaja comenzando con la última ecuación donde el residuo es diferente a cero y se reescribe en función al residuo, de la siguiente forma:

$$
\begin{array}{ccc}
22 = 3 \times 7 + 1 & \text{entonces} & 1 = 22 - (3 \times 7) \\
29 = 1 \times 22 + 7 & \text{entonces} & 7 = 29 - (1 \times 22) \\
660 = 22 \times 29 + 22 & \text{entonces} & 22 = 660 - (22 \times 29)
\end{array}
$$

Se reemplazan los valores 7 y 22 en $1 = 22 - (3 \times 7)$ obteniendo:

$$1 = [660 - 22 \times 29] - (3 \times [29 - 1 \times [660 - (22 \times 29)]])$$

$$1 = [660 - 22 \times 29] - (3 \times [29 - 660 + 22 \times 29])$$

$$1 = 660 - 22 \times 29 - 3 \times 29 + 3 \times 660 - 3 \times 22 \times 29)$$

$$1 = 660 + 3 \times 660 - 22 \times 29 - 3 \times 29 - 66 \times 29)$$

$$1 = (4) \times 660 + (-91) \times 29$$

Finalmente, s=4 y t=-91, por lo que se comprueba que:

$$\text{MCD}(660, 29) = 1 = 4 \times 660 + (-91) \times 29$$

Algoritmo 2: MCD(x, y)

entrada: x y y (enteros no negativos, ambos diferentes a cero)
salida : máximo común divisor de x y y
`/* sea` x `el mayor */`
if $x < y$ **then** intercambia(x,y) ;
while $y \neq 0$ **do**
    $r = x \bmod y$;
    $x = y$;
    $y = r$;
return a;

6. Mínimo Común Múltiplo

Si x, y y k están en $\mathbb{Z}^+$, y $x|k$, $b|k$, se dice que k es un **múltiplo común** de x y y. A la k más pequeña de éstas, a la que se llamará w, se la denomina **mínimo común múltiplo** o MCM, de x y y y se escribe w=MCM(x, y). El MCM se puede obtener a partir del máximo común divisor.

Sean x y y dos enteros positivos, entonces MCD(x, y) $\times$ MCM(x, y) = $x \times y$. Entonces

$$\text{MCM}(x, y) = \frac{x \times y}{\text{MCD}(x, y)}$$

Ejemplo:

1. Sean $x = 540$ y $y = 504$, el MCM(x, y) es:

$$\text{MCM}(x, y) = \frac{540 \times 504}{\text{MCD}(540, 504)} = \frac{272160}{36} = 7560$$

2. Sean $x = 30$ y $y = 105$

$$\text{MCM}(x, y) = \frac{30 \times 105}{\text{MCD}(30, 105)} = \frac{3150}{15} = 210$$

Si tenemos a $x = 30$ y $y = 105$, podemos encontrar todos los números primos que son factores de x y y que son: $p_1 = 2$, $p_2 = 3$, $p_3 = 5$ y $p_4 = 7$. Luego $x = 2^1 \times 3^1 \times 5^1 \times 7^0$, y $y = 2^0 \times 3^1 \times 5^1 \times 7^1$. Entonces se tiene:

$$MCD(30, 105) = 2^{\min(1,0)} \times 3^{\min(1,1)} \times 5^{\min(1,1)} \times 7^{\min(0,1)} = 2^0 \times 3^1 \times 5^1 \times 7^0 = 15$$

De la misma forma

$$MCM(30, 105) = 2^{\text{máx}(1,0)} \times 3^{\text{máx}(1,1)} \times 5^{\text{máx}(1,1)} \times 7^{\text{máx}(0,1)} = 2^1 \times 3^1 \times 5^1 \times 7^1 = 210$$

Ejercicios

Determine el MCD y MCM de:

1. 53 y 77
2. 231 y 1820
3. 82320, 950796

7. El Sistema Criptográfico de llave pública RSA

La **criptología** estudia a **sistemas** conocidos como **criptográficos** o **criptosistemas**, usadas para mantener las comunicaciones seguras. En este tipo de sistemas, el remitente transforma un mensaje antes de transmitirlo, esperando que sólo los receptores autorizados puedan reconstruir el mensaje original (antes de que el mensaje fuera transformado). Así, el remitente envía un mensaje **encriptado** o **cifrado**, y el receptor **desencripta** o **descifra** el mensaje. Con esto, las personas no autorizadas no podrán descubrir la técnica para encriptar y si lee el mensaje cifrado, no podrá descifrarlo. Se espera que se use este tipo de sistemas en organizaciones que deban proteger los datos del usuario, sea en la milicia, gobiernos, negocios relacionados a finanzas, e incluso conversaciones entre usuarios de una red social. Por ejemplo, si se envía el número de una tarjeta de crédito por internet, es importante que sólo el receptor al que se dirige pueda leerlo.

Antiguos sistemas tenías métodos sencillos, en los cuales el remitente y el receptor poseen una clave que define un carácter que sustituye cada carácter potencial a ser enviado, en estos sistemas las personas no revelan la clave, así esta clave se considera como *clave privada.*

Por ejemplo, si una llave se define como:

el carácter:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
se sustituye por:	EIJFUAXVHWP GSRKOBTQYDMLZNC

Si se desea enviar el mensaje ENVÍA DINERO, este sería cifrado como: ARMWIEUWRATK, del mismo modo, el mensaje cifrado UWRATKEARMWIUK se descifraría como: DINERO ENVIADO.

En este sistema, cada participante hace pública una llave de ciframiento y oculta la llave de desciframiento, así para enviar un mensaje, basta con buscar la llave de ciframiento en la tabla pública distribuida y el receptor descifra el mensaje usando la llave de desciframiento oculta. Vale indicar que en estos sistemas la llave se puede averiguar con facilidad, pues ciertas letras y combinación de letras aparecen con más frecuencia que otras, además la llave privada debe enviarse por un medio seguro antes de enviar el mensaje.

Otro sistema criptográfico conocido es el Sistema criptográfico RSA de llave pública, se llama así en honor de sus inventores Ronald L. Rivest, Adi Shamir y Leonard M. Adleman, que se piensa que es seguro. En este sistema, los mensajes se representan como números, por ejemplo, si un espacio en blanco se presenta como 1, A como 2, B como 3, etc. Por ejemplo:

El mensaje ENVIA DINERO se representaría como 6, 15, 23, 10, 2, 1, 5, 10, 15, 6, 19, 16. Y para combinarlo en un solo entero, se agregan ceros a la izquierda en todos los números de un sólo dígito, quedando: 061523100201051015061916.

7.1. Cálculo de potencias mod z

Para calcular potencias mod z se estudiará un algoritmo para calcular una potencia a^n (sin incluir mod z). Esta potencia se calcula multiplicando repetidas veces a por si mismo, n-a's, que usa $n-1$ multiplicaciones. Sin embargo, se logran mejores resultados si se **eleva al cuadrado repetidas veces**.

Por ejemplo, para calcular a^{29} se calcula $a^2 = a \times a$, que requiere de una multiplicación. Luego se calcula $a^4 = a^2 \times a^2$, que usa una multiplicación más. A continuación, se calcula $a^8 = a^4 \times a^4$, que usa una multiplicación adicional y después se calcula $a^{16} = a^8 \times a^8$, que requiere una multiplicación más. Hasta ahora se han empleado sólo 4 multiplicaciones, además, se observa que el número 29 en potencias de 2 (expansión binaria) se tiene que 29=16+8+4+1, por lo que se puede calcular como:

$$a^{29} = a^{16} \times a^8 \times a^4 \times a^1$$

que usa tres multiplicaciones adicionales, con un total de 7 multiplicaciones en lugar de 28 multiplicaciones necesarias por la forma tradicional.

A continuación se muestra cómo se calcula a^{29} elevando al cuadrado una y otra vez. Para iniciar $x = a$ y $n = 29$, se calcula n mod 2. Se coloca el resultado y se calcula el cociente de n actual dividido entre dos, luego el valor de n se actualiza con el cociente recién calculado, este proceso continúa hasta que el cociente sea cero.

x	Valor de n	n mod 2	Resultado	$n/2$
a	29	29 mod 2 = 1	a	14
a^2	14	14 mod 2 = 0	Sin cambio	7
a^4	7	7 mod 2 = 1	$a \times a^4$	3
a^8	3	3 mod 2 = 1	$a^5 \times a^8$	1
a^{16}	1	1 mod 2 = 1	$a^{13} \times a^{16}$	0

Algoritmo 3: *exp_via_cuadrado_repetido*(a, n)

entrada: a y n
salida : a^n
$resultado = 1$;
$x = a$;
while $(n > 0)$ **do**
  if $(n \ mod \ 2 == 1)$ **then** $resultado = resultado \times x$;
  $x = x \times x$;
  $n = \lfloor n/2 \rfloor$;
return *resultado*;

El algoritmo mostrado indica cómo funciona el cálculo de a^n, en el cual el ciclo termina cuando $n = 0$. Este algoritmo depende del tamaño de los números implicados. Sin embargo, si se solicita a^n mod z para valores grandes de a y n, el cálculo de a^n será enorme e impráctico, sobre todo si se desea calcular el residuo cuando a^n se divide entre z. Una forma de obtener mejores resultados tiene como idea clave el cálculo del residuo luego de cada multiplicación y así mantener los números relativamente pequeños.

$$ab \bmod z = [(a \bmod z)(b \bmod z)] mod z$$

Por ejemplo, para a^{29} se tiene:

$$a, \qquad a^5 = a \times a^4, \qquad a^{13} = a^5 \times a^8, \qquad a^{29} = a^{13} \times a^{16}$$

Entonces, para obtener a^{29} mod z, se calcula de manera sucesiva:

$$a \bmod z, \qquad a^5 \bmod z, \qquad a^{13} = \bmod z, \qquad a^{29} \bmod z$$

Se calcula de la forma:

$$
\begin{array}{rcrl}
a^2 \bmod z & = & a \times a \bmod z & = [(a \bmod z)(a \bmod z)] \bmod z \\
a^4 \bmod z & = & a^2 \times a^2 \bmod z & = [(a^2 \bmod z)(a^2 \bmod z)] \bmod z \\
a^8 \bmod z & = & a^4 \times a^4 \bmod z & = [(a^4 \bmod z)(a^4 \bmod z)] \bmod z \\
a^{16} \bmod z & = & a^8 \times a^8 \bmod z & = [(a^8 \bmod z)(a^8 \bmod z)] \bmod z \\
a^5 \bmod z & = & a \times a^4 \bmod z & = [(a \bmod z)(a^4 \bmod z)] \bmod z \\
a^{13} \bmod z & = & a^5 \times a^8 \bmod z & = [(a^5 \bmod z)(a^8 \bmod z)] \bmod z \\
a^{29} \bmod z & = & a^{13} \times a^{16} \bmod z & = [(a^{13} \bmod z)(a^{16} \bmod z)] \bmod z
\end{array}
$$

Por ejemplo, si se quiere calcular $572^{29} \bmod 713$, se debe considerar que el 572^{29} tiene 80 dígitos, por lo que sería necesario simplificar los cálculos como sigue:

$$
\begin{array}{lll}
572^2 \bmod 713 & = [(572 \bmod 713)(572 \bmod 713)] \bmod 713 & = (572 \times 572) \bmod 713 = 630 \\
572^4 \bmod 713 & = [(572^2 \bmod 713)(572^2 \bmod 713)] \bmod 713 & = (630 \times 630) \bmod 713 = 472 \\
572^8 \bmod 713 & = [(572^4 \bmod 713)(572^4 \bmod 713)] \bmod 713 & = (472 \times 472) \bmod 713 = 328 \\
572^{16} \bmod 713 & = [(572^8 \bmod 713)(572^8 \bmod 713)] \bmod 713 & = (328 \times 328) \bmod 713 = 634 \\
572^5 \bmod 713 & = [(572 \bmod 713)(572^4 \bmod 713)] \bmod 713 & = (572 \times 472) \bmod 713 = 470 \\
572^{13} \bmod 713 & = [(572^5 \bmod 713)(572^8 \bmod 713)] \bmod 713 & = (470 \times 328) \bmod 713 = 152 \\
572^{29} \bmod 713 & = [(572^{13} \bmod 713)(572^{16} \bmod 713)] \bmod 713 & = (152 \times 634) \bmod 713 = 113
\end{array}
$$

Esto se formaliza en el algoritmo siguiente, donde los números multiplicados son los residuos después de dividir entre z, por lo que tienen magnitud menor que z.

Algoritmo 4: *exp_mod_z_via_cuadrado_repetido*(a, n, z)

entrada: a, n y z
salida : $a^n \bmod z$
$resultado = 1$;
$x = a \bmod z$;
while $(n > 0)$ **do**
  if $(n \ mod \ 2 == 1)$ **then** $resultado = (resultado \times x) \bmod z$;
  $x = (x \times x) \bmod z$;
  $n = \lfloor n/2 \rfloor$;
return *resultado*;

7.2. Cálculo del Inverso del módulo de un entero

Para usar el sistema RSA se requiere calcular el **inverso de** n **mod** ϕ. Esto es, dados dos enteros $n > 0$ y $\phi > 1$ tal que MCD(n, ϕ)=1. Se calculará el valor de un entero s, $0 < s < \phi$ tal que $ns \bmod \phi = 1$ (para ello se encuentran los valores para $s'n + t'\phi = 1$) usando la fórmula:

$$s'n \bmod \phi = 1$$

Si s' fuera mayor a ϕ se encuentra el módulo respecto al s' encontrado, s sería:

$$s = s' \bmod \phi$$

Por ejemplo, si $n = 29$ y $\phi = 660$ se tiene que $s' = -91$ y $t' = 4$, entonces:

$$29(-91) \bmod 660 = ns' \bmod \phi = 1$$

Así, $s = s' \bmod \phi = -91 \bmod 660 = 569$. Por lo tanto, el inverso de 110 módulo 660 es 569.

7.3. Funcionamiento del sistema criptográfico RSA de llave pública

Primero, cada receptor potencial elige dos números primos p y q y calcula $z = pq$, la seguridad del sistema RSA se basa en la incapacidad de otros de conocer el valor de z, por lo que es común que los números p y q se elijan de manera que cada uno tenga 100 dígitos o más. Después, el receptor calcula $\phi = (p-1)(q-1)$ y elige un número entero n de tal forma que el MCD(n, ϕ)=1. Este valor n suele ser primo. El par z, n se hace público. Por último, el receptor potencial calcula el número único s, $0 < s < \phi$, que satisface ns mod $\phi = 1$. El número s se guarda en secreto y se usa para descifrar los mensajes.

Para enviar el entero a, $0 \leq a \leq z-1$, al propietario de la llave pública z, n, el remitente calcula $c = a^n$ mod z y envía c. Para descifrar el mensaje, el receptor calcula c^s mod z, que es igual a a.

Suponga que se elige $p = 23$, $q = 31$ y $n = 29$, entonces $z = pq = 23 \times 31 = 713$, $\phi = (p-1)(q-1) = (23-1)(31-1) = 22 \times 30 = 660$. Además $s = 569$ ya que se sabe que MCD(29,660)=1 y que ns mod $\phi = 29 \times 569$ mod $660 = 16501$ mod $660 = 1$. Luego se hace público $z = 713$ y $n = 29$.

Para transmitir $a = 572$ al dueño de la llave pública 713, 29, el remitente calcula $c = a^n$ mod z que es: $c = 572^{29}$ mod $713 = 113$. El receptor calcula c^s mod $z = 113^{569}$ mod $713 = 572$. El resultado principal que hace que funcione el ciframiento y desciframiento es que

$$a^n \bmod z = a, \text{ para todo } 0 \leq a < z \text{ y } u \bmod \phi = 1$$

Usando este resultado, se demuestra que el desciframiento produce el resultado correcto, como ns mod $\phi = 1$

$$c^s \bmod z = (a^n \bmod z)^s \bmod z = (a^n)^s \bmod z = a^{ns} \bmod z = a.$$

Actualmente no se cuenta con un algoritmo eficiente para factorizar enteros; es decir, no se conoce un algoritmo para factorizar enteros de d dígitos en tiempo polinomial, $O(d^k)$, ya que de esto depende la seguridad del sistema RSA. Además, si se seleccionan números primos p y q suficientemente grandes, sería impráctico calcular la factorización $z = pq$. Si una persona intercepta un mensaje, y pudiera encontrar la factorización, podría descifrar el mensaje igual que el receptor autorizado. Sin embargo, aún no se conoce un método práctico para factorizar enteros con 200 dígitos o más, de manera que si p y q se eligen cada uno con 100 dígitos o más, pq tendrá alrededor de 200 dígitos o más, lo que parece lograr que el sistema RSA sea seguro.

La primera descripción del sistema criptográfico RSA se encuentra en la columna de Matin Gardner en febrero de 1977 de Scientific American. Incluyendo un mensaje encriptado usando la clave z, n donde z era el producto de primos con 64 y 65 dígitos, y $n = 9007$, además de una oferta de \$100 a la primera persona que descifrara el código. Se estimaba que factorizar z tomaría unos 40 mil billones de años, sin embargo, en abril de 1994, Arjen Lenstra, Paul Leyland, Michaek Graff y Derek Atkins, factorizaron z con la ayuda de 600 voluntarios de 25 países, usando más de 1600 computadoras, coordinando el trabajo por internet.

Otra forma de interceptar y descifrar un mensaje sería tomar la raíz n de c mod z, donde c es el valor cifrado. Como $c = a^n$ mod z, la raíz n de c mod z daría a, el valor descifrado. Aún no se conoce ningún algoritmo que calcule las raíces n mod z en tiempo polinomial. También es posible que un mensaje se pueda descifrar por algún medio diferente a la factorización de enteros o la obtención de las raíces n mod z.

A mediados de los 90, Paul Kocher propuso una manera de penetrar en el RSA con base en el tiempo que toma descifrar un mensaje. La idea es que llaves secretas diferentes requieren tiempos diferentes para descifrar los mensajes y, al usar esta información de

tiempos, una persona no autorizada quizá sea capaz de descubrir la llave secreta y descifrar el mensaje. Para frustrar estos ataques, quienes están a cargo del RSA han tomado medidas para alterar el tiempo observado para descifrar los mensajes.

Ejercicios

Escriba un programa que:

1. Eleve a un exponente elevando al cuadrado repetidas veces (Algoritmo 3.)
2. Dados dos números enteros no negativos y diferentes de cero (a y b), calcule los enteros s y t que satisfacen MCD(a, b)=$sa + tb$
3. Dados dos números enteros $n > 0$ y $\phi > 1$, MCD(n, ϕ)=1, calcule el inverso de n mod ϕ.

8. Trabajos referidos al tema

1. Desarrolle los ejercicios propuestos en clase.
2. Realice los programas propuestos en clase.
3. Escriba con sus propias palabras un documento tipo monografía de mínimo 1 página y máximo 3 páginas (no es necesario que tenga carátula), indicando las referencias (páginas web, libros, etc.) que usó para realizar su trabajo de:
   a) Números primos.
   b) Criptografía.

Referencias

[1] Aguilar, A.S. (2009) Matemáticas para computación II. Universidad Estatal a Distancia, Escuela de Ciencias Exactas y Naturales.

[2] Grassman, W.K. & Tremblay J. (1997) Matemática Discreta y Lógica. Prentice Hall.

[3] Grimaldi, R.P. (2003) Discrete and Combinatorial Mathematics An Applied Introduction. Fifth Edition, Pearson, Prentice Hall.

[4] Jiménez, J. A. (2005) Matemáticas para la computación. Primera Edición, Alfaomega Grupo Editor, S.A. de C.V., México.

[5] Johnsonbaugh, R. (2005) Matemáticas Discretas, Sexta Edición, Pearson, Prentice Hall.

[6] Kolman, B., Busby, R.C. & Ross, S. (1995) Estructuras de Matemáticas Discretas para la Computación. Tercera Edición, Pearson, Prentice Hall.

[7] Miranda, F.E. & Viso, G.E. (2022) Matemáticas Discretas. Facultad de Ciencias, UNAM.

[8] Rosen, K. H., Michaels, J.G., Gross, J.L., Grossman, J.W. & Shier, D.R. (2000) Handbook of Discrete and Combinatorial Mathematics. CRC Press.