INICIACIÓN A LA INVESTIGACIÓN
REVISTA ELECTRÓNICA
UNIVERSIDAD DE JAÉN
Ini Inv, e1: a16 (2006)

Divisibilidad de los números enteros: El "secreto" de los números primos. (Una experiencia práctica de clase)

Miguel A. García-Muñoz, Carmen Ordóñez y Juan F. Ruiz

Departamento de Matemáticas (Área de Álgebra). Universidad de Jaén.
Campus Las Lagunillas s/n, 23071 Jaén.
jfruiz@ujaen.es

PRESENTACIÓN

Las matemáticas y en particular la matemática discreta es una materia abstracta, tradicionalmente considerada como compleja y bastante novedosa para el alumnado; la monotonía de los contenidos, el abandono durante el curso de la asignatura, la falta de motivación e interés, el poco o ningún conocimiento de contenidos previos, los prejuicios de los alumnos, son los principales problemas a los que se enfrenta el docente. Ajustándonos a un tema particular, mostramos como podemos acercar al alumno estos contenidos; a modo de ejemplo, y muy en concreto, basándonos en un concepto sobradamente conocido por todos como el de número primo; en general, no suele quedar patente la importancia de éste en la vida cotidiana y tampoco su uso generalizado, especialmente en temas relacionados con la informática. Nos valemos de ello para motivar de manera efectiva al alumnado (especialmente a un alumnado de la Ingeniería de Informática de Gestión).

OBJETIVOS

Los alumnos de informática muestran por lo general un natural interés y curiosidad por todos los temas relacionados con las nuevas tecnologías, trasladar los abstractos conceptos teóricos a un ambiente familiar para este tipo de alumnado, supone la revitalización del interés del mismo por la asignatura y dota al docente de mayor capacidad de enseñanza. Aquí mostramos un ejemplo centrado en el tema de la divisibilidad de números enteros y los números primos que ilustra estas ideas. Nuestros principales objetivos son: *traducir* al lenguaje informático (lenguaje de programación) los principales conceptos teóricos; *demostrar* al alumno la utilidad de los contenidos propuestos y *motivar* e *implicar* al alumno en un seguimiento efectivo de los contenidos de la asignatura.

DESARROLLO

En esta experiencia práctica correspondiente a la asignatura de Álgebra I de Informática de Gestión, mostramos un ejemplo de como se puede llevar a cabo lo antes expuesto.

A. PRELIMINARES TEÓRICOS

Nos encontramos estudiando la divisibilidad de números enteros que introducimos de forma teórica como sigue:

Dados a, $b \in Z$, diremos que <u>a divide a b</u>, <u>a es divisor de b</u> o <u>b múltiplo de a</u>:

$$a \mid b \text{ si y sólo si } \exists\, c \in Z \text{ tal que } b = ac$$

Diremos que un <u>número</u> entero p es <u>primo</u>, si $p \neq 0, 1, -1$ y sus únicos divisores son p, $-p$, 1, -1. La importancia de los primos se pone de manifiesto en el siguiente resultado:

Teorema. Teorema Fundamental de la Aritmética. Todo número entero n, distinto de 0, 1, -1, se escribe de forma única (salvo el orden y el signo) como producto de números enteros primos.

$$n = \pm p_1^{e_1} \ldots\ldots p_r^{e_r}$$

En el estudio de la divisibilidad en números enteros, se aborda el concepto de congruencia:

Sean a, $b \in Z$ y $n \in N$. Diremos que <u>a es congruente con b módulo n</u> , y lo notaremos, $a \equiv b \bmod n$, si y sólo si $a - b$ es múltiplo de n. Es decir:

$$a \equiv b \bmod n \Leftrightarrow \exists\, k \in Z \text{ tal que } a = b + kn$$

Fácilmente se puede comprobar que ésta es una relación de equivalencia en Z, cuyo cociente son las clases de restos módulo n, que conocemos como Z_n . Así es obvio que en Z_n ,

$$\bar{a} = \bar{b} \text{ sii } a \equiv b \bmod n$$

Una vez que se ha definido el conjunto Z_n, se definen en él dos operaciones, suma y producto que dotan a Z_n de una estructura algebraica de anillo conmutativo, en particular si n es primo se comprueba que Z_n es un cuerpo:

Sabemos que Z_n es, en general, un anillo conmutativo. Sus unidades (es decir los elementos que admiten inverso) son fácilmente identificables; en efecto,

$$\bar{a} \text{ admite inverso si y sólo si } (a,n) = 1$$

Además el cálculo del inverso puede obtenerse como una consecuencia del algoritmo de Euclides. Concretamente, si $\bar{a}$ admite inverso es porque $(a,n) = 1$, y por la Identidad de Bezout, $\exists\, u, v \in Z$ tales que $1 = au + nv$. Tomando clases en la igualdad anterior, tendremos $\bar{1} = \bar{a}\bar{u}$ pues $\bar{n} = \bar{0}$. Así deducimos que el inverso de $\bar{a}$ es la clase $\bar{u}$ siendo u el número que aparece en la Identidad de Bezout acompañando al entero a.

Es evidente el nivel de abstracción implícito en cualquier concepto teórico y de forma particular en los expuestos anteriormente. También es clara la monotonía que sobrelleva el uso reiterado del lenguaje matemático, por otra parte

indispensable para mostrar de forma objetiva y concreta los conceptos necesarios.

B. HERRAMIENTAS INFORMÁTICAS

Después de la base teórica y de realizar distintos ejemplos en la pizarra, se les proporcionan las herramientas informáticas necesarias para realizar los mismos problemas de clase con el ordenador. La finalidad es doble, por un lado proporcionar al alumnado una herramienta que le permita comprobar la fidelidad de los resultados obtenidos, por otro lado una traducción de los mismos conceptos matemáticos a un lenguaje informático, supuestamente familiar para un estudiante de informática. En la experiencia que aquí contamos se necesitan dos programas, ambos están implementados con el programa Mathematica[1], aunque son fácilmente transportables a cualquier lenguaje de programación:

1. El algoritmo de Euclides.

```
n1= ENTRADA DEL ALGORITMO;
n2= ENTRADA DEL ALGORITMO;
a=Abs[n1];b=Abs[n2];
If [a<b,a=b;b=Abs[n1];];
m=1;
While[m>0,m=Mod[a,b];a=b;b=m;];
Print["m.c.d.(",n1,",",n2,")=",a]
Print["m.c.m.(",n1,",",n2,")=",Abs[(n1*n2)/a]]
```

2. La identidad de Bezout.

```
Clear[valor1,valor2];
n1= ENTRADA DEL ALGORITMO;
n2= ENTRADA DEL ALGORITMO;
If[Abs[n1]>Abs[n2], temp=n1; n1=n2; n2=temp];
Signo1=n1/Abs[n1];Signo2=n2/Abs[n2];
a=Abs[n1];b=Abs[n2];
If[a<b,a=b;b=Abs[n1];n3=a;n4=b;];
If [Mod[n1,n2]==0, Valoru=0;Valordev=Signo1; a=b;,
r=1;Cocientes={};s=0;
While[r>0,
   q=Quotient[a,b];r=Mod[a,b];a=b;b=r;s=s+1;
   AppendTo[cocientes,q];
];
Listam=Table[0,{i,s}];
Listam[[1]]=valor1;Listam[[2]]=valor2;
For [f=3,f<s+1,f++,
  Listam[[f]]=listam[[f-2]]-(listam[[f-1]]*cocientes[[f-2]])
];
Bezout:=Simplify[listam[[s-1]]-(listam[[s]]*cocientes[[s-1]])];
valor1=1;valor2=0;
Valoru=Bezout;
valor1=0;valor2=1;
Valordev=Bezout;
Valoru=Valoru*Signo2;
Valordev=Valordev*Signo1;
Print["m.c.d.{",n1,",",n2,"}=",a]
Print["m.c.m.{",n1,",",n2,"}=",(n3*n4)/a]
Print["Identidad de Bezout: ",a," = ",n2,"·(",Valoru,") +
",n1,"·(",Valordev,")."]
```

[1] El programa Mathematica es una aplicación comercial de Wolfram Research especialmente diseñada para realizar cálculos y operaciones matemáticas.

C. EXPERIENCIA PRÁCTICA

No es posible evitar el lenguaje matemático u omitir tales conceptos, podemos como se muestra en el apartado 3 trasladarlos fielmente a un lenguaje informático, sin embargo, y aunque el uso del ordenador provoca un aumento del interés, el nivel de abstracción continua siendo elevado y en consecuencia, buena parte del alumnado continúa sin incorporarse al seguimiento de la materia, para evitarlo podemos rebajar los conceptos y motivar al alumno con algunos ejemplos relacionados con el tema que tratamos, muy cercanos al público en general y de forma particular al alumno de informática:

Ejemplo 1. La letra del DNI español no es más que un código de control que comprueba si el DNI es correcto. Este puede determinarse fácilmente calculando el resto de dividir el DNI entre 23 y asignarle a cada resto una letra según la siguiente tabla:

RESTO	0	1	2	3	4	5	6	7	8	9	10	11
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B

RESTO	12	13	14	15	16	17	18	19	20	21	22	
LETRA	N	J	Z	S	Q	V	H	L	C	K	E	

$$DNI \equiv RESTO \bmod 23$$

Con el ordenador:

```
Letras={"T","R","W","A","G","M","Y","F","P","D",
        "X","B","N","J","Z","S","Q","V","H","L",
        "C","K","E"};
dni=NÚMERO_DNI;
Letras[[Mod[dni,23]+1]]
```

Ejemplo 2. Un número de cuenta bancario consta de 20 dígitos: los cuatro primeros indican la entidad, los cuatro siguientes indican la oficina, los dos siguientes son dígitos de control y los diez últimos se corresponden con el número de cuenta personal. Los dos dígitos de control se obtienen, el primero de los ocho dígitos de la entidad y oficina y el segundo del número de cuenta, de forma que son un test para verificar que el número es correcto. Llamemos D al primer dígito de control y C al segundo.

---

[2] Comprobemos que en efecto C^d es congruente modulo n al mensaje original. Es claro que $C^d = (M^e + kn)^d$ en Z_n.
Y como en el desarrollo del binomio el único sumando que no es múltiplo de n es $M^{(ed)}$, entonces,

$$C^d = (M^e + kn)^d = C^d = M^{(ed)} \text{ en } Z_n.$$

Para terminar bastaría con comprobar que $M^{(ed)} \equiv M \bmod n$. Comprobar esto último no es inmediato y para ello necesitamos de la Identidad de Euler-Fermat, sin entrar en detalles diremos que para un primo p se verifica $M^{p-1} \equiv 1 \bmod p$, que es lo que necesitamos, en nuestro caso sabemos que $ed \equiv 1 \bmod (p-1)(q-1)$, luego, $M^{(ed)} = M^{((p-1)(q-1)k+1)} = (M^{p-1})^{(q-1)k}M \equiv M \bmod p$, y análogamente podemos razonar para q, entonces $M^{(ed)} \equiv M \bmod q$, y como p y q son primos entonces $M^{(ed)} \equiv M \bmod pq$.

[3] Uno de los números primos más grandes es $2^{20996011} - 1$ con más de seis millones de dígitos, que fue descubierto el 17 de noviembre de 2003.

Entidad (4)	Oficina (4)	DC (2)	N°Cuenta (10)
$E_1\ E_2\ E_3\ E_4$	$E_5\ E_6\ E_7\ E_8$	D C	$N_1\ N_2\ N_3\ N_4\ N_5\ N_6\ N_7\ N_8\ N_9\ N_{10}$

Veamos como se calculan los dígitos de control, consideremos,

$$\{d_i\}_{i=1,2,\ldots,8} = \{4,8,5,10,9,7,3,6\}$$
$$\{c_i\}_{i=1,2,\ldots,10} = \{1,2,4,8,5,10,9,7,3,6\}$$

Ahora calculamos $M = \sum_{i=1}^{8} d_i E_i$ y $N = \sum_{i=1}^{10} c_i N_i$, y por último:

$$D \equiv \begin{cases} M \bmod 11, & si\ M \equiv 0,1 \bmod 11. \\ -M \bmod 11, & en\ otro\ caso. \end{cases}$$
$$C \equiv \begin{cases} N \bmod 11, & si\ N \equiv 0,1 \bmod 11. \\ -N \bmod 11, & en\ otro\ caso. \end{cases}$$

Con el ordenador:

```
d={4,8,5,10,9,7,3,6};c={1,2,4,8,5,10,9,7,3,6};
EntidadOficina= {E1,E2,E3,E4,E5,E6,E7,E8};
Cuenta= {N1,N2,N3,N4,N5,N6,N7,N8,N9,N10};
m=0;n=0;
Do[m=m+(d[[i]]*EntidadOficina[[i]]),{i,1,8}];
Do[n=n+(c[[i]]*Cuenta[[i]]),{i,1,10}];
digitoD=If[Mod[m,11]<2, Mod[m,11],11-Mod[m,11]]
digitoC=If[Mod[n,11]<2, Mod[n,11],11-Mod[n,11]]
```

El primer ejemplo muestra un uso directo y cotidiano de las congruencias, el segundo además muestra como se puede usar Z_{11}. Ambos ejemplos juegan con la curiosidad por conocer algo de uso habitual para motivar al alumno e interesarlo en la asignatura.

Aunque es evidente el provecho de los ejemplos anteriores, para motivar de forma efectiva al alumno, ambos ejemplos usan de manera superficial la teoría, por tanto, es necesario plantearle retos más complejos, que tengan también un uso cotidiano y despierten aún más su curiosidad. Para ello se les puede plantear un nuevo ejemplo, esta vez, aunque de mayor dificultad, de aplicación muy directa y rápida para que el alumno no se pierda en entelequias lógicas y que les implica en un trabajo personal, pero muy relacionado con su titulación. Un problema adecuado para el tema particular que tratamos podría ser el siguiente:

Ejemplo 3. Uno de los algoritmos de encriptación más conocidos y usados es el algoritmo de Rivest-Shamir-Adleman, más conocido como RSA en honor a sus autores que lo crearon en 1978. Pasamos a describirlo, tan sólo necesitamos de dos números primos p y q (generalmente muy grandes, más de 100 cifras, esto es mayor que 10^{100}). Con ellos calculamos:

1. En primer lugar n = pq.
2. Calculamos z = (p - 1)(q - 1).
3. Buscamos un entero positivo e < n tal que (e, z) = 1.

4. Como e y z son primos relativos, existe el inverso de e módulo z, lo llamamos $d \equiv e^{-1} \bmod z$. (Además salvo múltiplo de z es único).

Realizados los cálculos; p, q y z no son necesarios para la encriptación del mensaje y su posterior descifrado. Tan sólo necesitamos de n, d y e. La codificación del mensaje se realizará por bloques, a cada uno de ellos se le asigna un número entero entre 0 y n-1 (salvo que la unicidad no sea importante) o la correspondiente cadena de ceros y unos si usamos el sistema de numeración binario, luego para un entero M, la codificación sería:

$$C \equiv M^e \bmod n$$

Y el descifrado[2],

$$M \equiv C^d \bmod n$$

Los valores n y e son públicos, y sólo se oculta el valor de d para que la encriptación sea segura. Luego si conociéramos d, sabríamos descifrar el mensaje, sin embargo, d sólo puede conocerse si sabemos factorizar n y los algoritmos de factorización son lentos, por tanto, un algoritmo RSA es más seguro cuanto más grandes[3] sean los primos p y q que podemos encontrar. Por otra parte, nótese también que el cifrado y descifrado necesita calcular una potencia y el resto de una división.

a. Encontrar dos números primos de al menos 2 cifras y usarlos para determinar z, e y d.
b. Programar una rutina que cifre mensajes de texto a partir de los códigos ASCII y otra que los descifre; usando los datos del apartado a.

Con el ordenador, tomando p=31, q=97 y e=59:

```
p=31;q=97;e=59;
n=p*q
z=(p-1)*(q-1)
```

Con el algoritmo de Euclides se comprueba que e y z son primos relativos y con la Identidad de Bezout se calcula el inverso de e en Z_z, resultando:

```
d=2099
```

Introducimos el mensaje y nos quedamos con sus códigos ASCII,

```
mensaje="Esto es una prueba"
listado=Characters[mensaje];
Do[listado[[i]]=ToCharacterCode[listado[[i]]][[1]],
{i,1,Length[listado]}]
listado
```

Codificamos el mensaje con:

```
Do[listado[[i]]=Mod[listado[[i]]^e,n],
{i,1,Length[listado]}]
listado
```

Y lo decodificamos con:

```
Do[listado[[i]]=Mod[listado[[i]]^d,n],
 {i,1,Length[listado]}]
Listado
```

RESULTADOS

La seguridad de muchos sistemas cotidianos, como las firmas digitales, televisiones de pago, cotización de valores… se basan en este sistema de encriptación y por tanto en la existencia de números primos, cuanto más grandes más seguros. Nuestra información y nuestros secretos están confiados a los números primos.

La importancia de los números primos, de la factorización de números enteros, de las congruencias, queda patente en los ejemplos anteriores, de esta forma demostramos al alumno que se encuentra ante unos contenidos relevantes e importantes para su vida profesional, por otro lado también se juega con el lógico interés que despierta un sistema de cifrado que pueden encontrar en cualquier rincón de su vida cotidiana, además al utilizarse el mismo lenguaje simbólico que el usado en las clases teóricas, y al resolverse con las herramientas informáticas proporcionadas en las clases prácticas se consigue por una parte, la implicación del alumnado y por otra, la familiarización del mismo con los conceptos y el lenguaje que pretendemos enseñar.

CONCLUSIÓN

No cabe pensar que el uso de nuevas tecnologías en el aula y más aún en la titulación de informática pueda de ningún modo ser perjudicial. Por el contrario parece el camino a seguir en asignaturas abstractas y aparentemente lejanas de cualquier aplicación práctica, es preciso demostrar al alumnado su utilidad, para evitar la desmotivación del mismo, y dentro de lo posible, mostrarle como se resuelven problemas reales y cercanos que puedan encontrar en su futura vida profesional. La motivación del alumnado con este tipo de experiencias es incuestionable. El principal problema que plantean estas experiencias es el tiempo necesario para desarrollarlas en clase, por otra parte no todos los contenidos de la materia permiten experiencias tan prácticas como esta, aunque con el tiempo y la experiencia docente se pueden llegar a conseguir en temas aparentemente poco proclives.