Algoritmos

$$\frac{1280}{345} = 3 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{4 + \cfrac{1}{2}}}}}$$

Edición Octubre - 2013

Colección Hojamat.es

http://www.hojamat.es

PRESENTACIÓN

La utilidad mayor de una hoja de cálculo en el estudio de los números es la de implementación de algoritmos. Con ellos se logra automatismo y velocidad, liberando así nuestro tiempo para tareas más interesantes.

La programación de un algoritmo es una tarea altamente formativa y entretenida, aunque requiere algo de experiencia. En este documento usaremos el Basic para hojas de cálculo como codificación básica, pero podrán aparecer los algoritmos en pseudocódigo u organigramas, así como para otras herramientas.

Para dar más facilidades de comprensión, en algunos algoritmos incluiremos versiones para la calculadora WIRIS, el programa WxMaxima y el lenguaje PARI. Las tres herramientas son gratuitas y fácilmente descargables. En otras ocasiones usaremos nuestra calculadora StCalcu

http://hojamat.es/sindecimales/aritmetica/herramientas/herrarit.htm

Como advertiremos en todos los documentos de esta colección, el material presentado no contiene desarrollos sistemáticos, ni pretende ser un manual teórico. En cada tema se incluirán cuestiones curiosas o relacionadas con las hojas de cálculo, con la única pretensión de explicar algunos conceptos de forma amena.

TABLA DE CONTENIDO

ALGORITMOS GENERALES

PERIODO DE LA FRACCIÓN 77/23

Sacar decimales

Las hojas de cálculo están orientadas a los números decimales y se comportan mal en algunos problemas que necesitan operaciones con números enteros. Así, en la cuestión de obtener el periodo de una fracción, aunque es un problema propio de números racionales, los cálculos se efectúan mediante la división entera tradicional. Así se efectuaba en las aulas cuando no existían las calculadoras.

No es difícil implementar una división entera para obtener los periodos largos que se producen con denominadores que contengan como factores números primos grandes. La idea es usar las funciones COCIENTE y RESIDUO

Por ejemplo, para obtener muchas cifras decimales del cociente 77/23, podemos proceder así: El cociente entre ambos sería COCIENTE(77;23) = 3, que sería la parte entera. El resto se hallaría mediante RESIDUO(77;23)=8.

A continuación podemos imitar la división que efectuábamos en el colegio (“sacar decimales”). Podemos multiplicar el resto 8 por 10 y volver a repetir la operación: COCIENTE(80;23) = 3, que sería la primera cifra decimal. Volvemos a hallar el resto: RESIDUO(80;23) = 11. Y así reiteramos cuantas veces deseemos.

En la imagen puedes estudiar la forma de ordenar estos cálculos

Núm. cifras	Restos		Cifras del cociente	
0	8		3	Parte entera del cociente
1	11		3	
2	18		4	
3	19		7	
4	6		8	
5	14		2	
6	2		6	
7	20		0	
8	16		8	
9	22		6	
10	13		9	Cifras decimales
11	15		5	
12	12		6	
13	5		5	
14	4		2	
15	17		1	
16	9		7	
17	21		3	
18	3		9	
19	7		1	

Puedes estudiar este algoritmo en el archivo de dirección

http:www.hojamat.es/aritmetica/teoria/hojas/granperiod.ods

Técnicas parecidas hemos usado para encontrar el periodo y anteperiodo en forma de cadena de caracteres. Se puede consultar en

http://hojamat.es/sindecimales/aritmetica/herramientas/herrarit.htm#periodo

Otras codificaciones

En PARI

Este algoritmo tiene fácil implementación en PARI:

M=77;N=23;T=100;for(i=1,T,Q=M\N;R=M%N;print1(Q);M=R*10)

El código ha sido preparado para obtener 100 cifras del cociente 77/23

En Wiris

Se pueden cambiar los valores 77, 23 y 100 para adaptarlos a otros datos.

```
si..sino   locales   mientras

m=77 → 77
n=23 → 23
d=0 → 0
s=0 → 0
repetir          → 100
   q=coc(m,n)
   r=resto(m,n)
   s=s·10+q
   m=r·10
   d=d+1
hasta d=100
s → 3347826086956521739130434782608695652173913043478260869565217391304347826086956521739130434782608695
```

En wxMaxima

```
m:77$
n:23$
s:0$
for i:1 thru 100 do
(
q:floor(m/n),
r:m-q*n,
m:r*10,
s:s*10+q
)$
s;
3347826086956521739130434782608695652173913043478260869565217391304347826086956521739130434782608695
```

También se pueden cambiar los datos 77, 23 y 100

Función en Stcalcu

Admite como datos dos números escritos entre comillas o en una celda con formato de texto, así como el número de decimales escrito con formato numérico:

```
Function stdecimales$(a$, b$, num)
Dim p$, m$, q$
Dim i

p$ = stdivi(a$, b$)
p$ = p$ + "."
m$ = residuo$
For i = 1 To num
m$ = m$ + "0"
Next i
q$ = stdivi(m$, b$)
p$ = p$ + q$
stdecimales = p$
End Function
```

EL ALGORITMO DE MOESSNER

Presentamos en este apartado una curiosidad matemática a base de cribados: toma la lista de los primeros números naturales. Tacha después uno de cada cuatro, comenzando con el mismo 4:

1 2 3 5 6 7 9 10 11 13 14

Después escribe la lista de sus sumas parciales.

1 3 6 11 17 24 33 43 54 67 81

Y ahora tachas de tres en tres, sumando después de nuevo.

1 3 11 17 33 43 67 81

1 4 15 32 65 108 175 256

Después tachas de dos en dos

1 15 65 175

Y sumas

1 16 81 256

El resultado es la serie de las potencias cuartas de los naturales. Recuerda que hemos comenzando tachando de cuatro en cuatro. ¿Funcionará con el tres?

Lo escribimos sin explicaciones:

1 2 4 5 7 8 10 11 13 14 16 17

1 3 7 12 19 27 37 48 61 75 91 108

1 7 19 37 61 91

1 8 27 64 125 216

Resultan los cubos. Prueba de dos en dos y obtendrás los cuadrados. ¿Funcionará esto siempre? Este algoritmo lo propuso Alfred Moessner y fue demostrada su validez para cualquier valor natural por Oskar Perron en 1951 usando la inducción matemática.

Nuestro objetivo hoy es reproducir este algoritmo con hoja de cálculo, que por cierto no es nada fácil. Contiene una verdadera trampa, que es la posible confusión entre valores y posiciones. Lo vemos:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	A. Roldán 2011												
2													
3								Algoritmo de Moessner					
4													
5													
6													
7		1	2	3	4	5	6	7	8	9	10	11	12
8		0	0	0	1	1	1	2	2	2	3	3	3
9	**4**	1	2	3	5	6	7	9	10	11	13	14	15
10		1	3	6	11	17	24	33	43	54	67	81	96
11		0	0	1	1	2	2	3	3	4	4	5	5
12	**3**	1	3	11	17	33	43	67	81	113	131	171	193
13		1	4	15	32	65	108	175	256	369	500	671	864
14		0	1	2	3	4	5	6	7	8	9	10	11
15	**2**	1	15	65	175	369	671	1105	1695	2465	3439	4600	5800
16		1	16	81	256	625	1296	2401	4096	6561	10000	14600	20400
17													
18	**1**												
19													
20													
21													

En la celda A9 escribimos la amplitud de los saltos. En la imagen está preparado para que resulten las cuartas potencias. La hoja se encarga de ir restando una unidad hacia abajo y dejar de escribir cuando se llegue a 1. El modelo está preparado para llegar a 5, pero si lo descargas puedes ampliarlo a tu gusto.

La fila 7 contiene la serie de números naturales. Después se van repitiendo hacia abajo tres filas:

Primera: Es un artificio, pues la hoja debe buscar el elemento a tachar cada vez más lejos, y dependiendo del valor de A9. Esto lo hemos resuelto con la fórmula (usamos la contenida en C8)

=SI(ESNUMERO($A12);SI(RESIDUO(C$7-1;$A12)=0;B8+1;B8);"")

En primer lugar verifica si aún quedan saltos por dar con **ESNUMERO($A12**). Después encuentra el residuo del número de arriba respecto al salto y hace avanzar el contador (B8) si ese número es múltiplo del salto. Así medimos el alejamiento del elemento que debemos tachar. Observa que van aumentando los valores cada tres (representan los tres supervivientes después de tachar)

Segunda: Aquí se eligen los números entre los de arriba, saltando los que ocupan un lugar múltiplo de 3. Después, con la función DESREF se dirigen a la celda adecuada para copiar el número:

=SI(ESNUMERO($A12);DESREF(C9;-2;C8);"")

DESREF se dirige a dos filas más arriba (-2) y salta según indica el valor de arriba (C8). Como esta contiene los saltos adecuados, cada vez que cambie su valor se tacha un número. Es lo que queríamos. No es fácil de entender y cuesta encontrar el procedimiento.

Tercera: Se limita a acumular sumas, y al llegar al nivel 1 produce las potencias deseadas.

Aunque esto no pasa de una curiosidad, la construcción del algoritmo es apasionante. Este que ofrecemos no usa macros, y lo puedes descargar en dos versiones desde

hojamat.es/blog/moessner.zip

EL PROBLEMA DE HAMMING

Reciben el nombre de números regulares o 5-lisos aquellos números naturales que son divisibles entre 2,3 y 5 y ningún otro factor primo. Presentan una factorización prima del tipo $2^n3^m5^p$. También puedes identificarlos como aquellos que son divisores de una potencia de 60 (¿por qué?)

Los tienes presentados en estas páginas

http://en.wikipedia.org/wiki/Regular_number

https://oeis.org/A051037

En esta última puedes consultar cuáles son

1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, 36, 40, 45, 48,…

Si se les añade el número 1 como primer elemento, forman la llamada sucesión de Hamming.

El mayor interés que presentan estos números es el estudio de la formación ordenada de la sucesión, formándola **a partir de los elementos ya descubiertos**. Esto último es importante, pues si no, bastaría con ir recorriendo los números naturales para quedarnos sólo con los del tipo $2^n3^m5^p$.

Los posibles algoritmos, como el de Dijkstra, se estudian frecuentemente en programación funcional, como puedes ver en esta página de José_A_Alonso.

Nosotros, como siempre en este blog, optaremos por un enfoque elemental, didáctico y ¡cómo no!, usando una hoja de cálculo.

Generación del siguiente número de Hamming

Una vez que tienes escritos los primeros números de la sucesión 1,2,3,4,5,6,8,…(veremos que en realidad sólo hay que escribir 1), para obtener el siguiente bastará multiplicar uno de ellos por 2,3 o 5, pero, ¿cuál? En la sucesión 1,2,3,4,5,6,8,… no me sirve multiplicar por 2 el número 3, porque me daría 6 que ya lo tengo. Sí me convendría multiplicar por 2 el 5, con lo que obtendría el 10, o al revés, multiplicar 5 por 2. Esto no tiene nada de sistemático, por lo que deberemos ordenarlo un poco:

Dada una sucesión de Hamming con varios elementos, para formar el siguiente nos basaremos en estos criterios:

(1) Multiplicamos por 2, 3 o 5 todos los elementos que ya tenemos, y nos quedamos con el resultado menor que aún no esté incorporado a la sucesión. En el caso del ejemplo, el 9.

(2) Para guiarnos en este proceso, escribimos todos los números del tipo 2H, 3H y 5H, (representando H los números de Hamming que ya tenemos) en tres columnas.

H	2H	3H	5H
1	2	3	5
2	4	6	10
3	6	9	15
4	8	12	20
5	10	15	25
6	12	18	30
8	16	24	40

(3) En cada columna señalamos aquel número que cumple que todos los de arriba no superan el número que ya tenemos (8) y él es el primero que sí lo sobrepasaría.

En la siguiente tabla los tienes señalados en este caso.

H	2H	3H	5H
1	2	3	5
2	4	6	**10**
3	6	**9**	15
4	8	12	20
5	**10**	15	25
6	12	18	30
8	16	24	40

(4) Por último, de los tres candidatos elegimos el menor, 9, y ese será el siguiente elemento de la sucesión de Hamming.

Reiteramos estas operaciones y los obtendremos todos de forma ordenada. Este procedimiento tiene la ventaja de que una vez elegido un número de la columna quedan desechados los anteriores, por lo que es posible mantener unos punteros que nos indiquen por dónde vamos.

El algoritmo con hoja de cálculo

Podemos traducirlo a hoja de cálculo. Lo hemos intentado sin usar macros, pero aparecían referencias circulares muy molestas, por lo que hemos acudido al uso de rutinas y botones. Se inicia el proceso con el botón Inicio, que escribe el primer término 1 y sus tres múltiplos 2,3 y 5.

Después, cada vez que pulsemos sobre el botón Paso se irán eligiendo los múltiplos adecuados desechando los anteriores y los iguales. En la imagen puedes ver el estado del proceso después de obtener el 9:

	Inicio		Paso
Hamming			
1			
2			10
3			15
4		12	20
5	10	15	25
6	12	18	30
8	16	24	40
9	18	27	45

Se han dejado en blanco los múltiplos usados. La hoja elige después el mínimo (sería 10), elimina sus iguales, lo incorpora a la lista, crea sus múltiplos y borra los innecesarios.

	Inicio		Paso
Hamming			
1			
2			
3			15
4		12	20
5		15	25
6	12	18	30
8	16	24	40
9	18	27	45
10	20	30	50

El cómo lo consigue lo podrás estudiar descargando la hoja en Excel desde

http://hojamat.es/blog/hamming.xlsm

Estudio mediante funciones

Para ver si un número es regular o 5-liso bastaría con esta definición de función:

```
Public Function es_regular(n) As Boolean
Dim nn

nn = n
While nn = 2 * Int(nn / 2): nn = nn / 2: Wend
While nn = 3 * Int(nn / 3): nn = nn / 3: Wend
While nn = 5 * Int(nn / 5): nn = nn / 5: Wend
If nn = 1 Then es_regular = True Else es_regular = False
End Function
```

Observa cómo lo detecta: mientras el número sea par, lo va dividiendo entre 2, con lo que al final deja de serlo. Mientras sea múltiplo de 3 y de 5 también va dividiendo. Si el número es regular se agotarán todos los factores y quedará sólo un 1 y el valor de la función será VERDADERO. Si no es regular es porque o no se puede dividir entre 2,3 o 5, o al final del proceso queda un factor mayor que 1, y la función devuelve FALSO.

Con esta función puedes iniciar la sucesión de Hamming en el punto que desees. Basta ir recorriendo números y eligiendo los que sean regulares. También es muy sencillo usar la función ***proximo_regular***:

```
Public Function proximo_regular(n)
Dim p

p = n + 1
While Not es_regular(p): p = p + 1: Wend
proximo_regular = p
End Function
```

Con esta función puedes descubrir, por ejemplo, que el primer regular de siete cifras es 1012500.

ALGORITMO 196

En el blog “Espejo lúdico”, con fecha 9 de Diciembre de 2008, se ha publicado esta propuesta:

Si a un número se le suma su reverso (por ejemplo 75 + 57) y se hace lo mismo con el resultado, llega un momento en que el resultado es capicúa.

Por ejemplo

75 + 57 =132; 132 +231 = 363

Se llama así el algoritmo porque para el número 196, en el momento de escribir este texto, aún no se sabe si el proceso para en un capicúa concreto o se prolonga indefinidamente. Se han llegado a organizar búsquedas que han durado años.

El 196 es el primero de los llamados números de Lychrel, de los que aún no se sabe si producen una parada en el algoritmo. Los siguientes son 295, 394, 493, 592, …(http://oeis.org/A023108). Todos están cerca de un número terminado en cero.

Para un cierto número de dos cifras es necesario repetir este proceso más de 10 veces. ¿Cuál es ese número?

El algoritmo propuesto tiene fácil ejecución para quien sepa sumar, pero no es tan simple para una hoja de cálculo. Hay que tener en cuenta que los números se almacenan en formato binario y la hoja “no sabe” la cifras que tiene un número en el sistema de numeración decimal. Por tanto, si no definimos nuevas funciones, no podrá invertir las cifras de un número ni tampoco averiguar si es capicúa o no. Así que necesitamos:

Función INVERTIR_CIFRAS: Debe de actuar sobre un número e invertir el orden de todas sus cifras, devolviéndonos el resultado.

```
Public Function cifrainver(n)
Dim l, i
Dim c
Dim auxi$, auxi2$, ci$

  ' invierte el orden de las cifras para dar otro número

auxi = Right(nn$, Len(nn$) - 1)
auxi2$ = ""
l = Len(auxi)
For i = l To 1 Step -1
ci$ = Mid(auxi, i, 1)
auxi2 = auxi2 + ci$
Next i
c = Val(auxi2$)
cifrainver = c
End Function
```

Función ESCAPICUA: Debe averiguar si un número es capicúa o no y devolver VERDADERO o FALSO

```
Public Function escapicua(n) As Boolean
Dim l, i, k
Dim c As Boolean
Dim auxi$

auxi = Right(nn$, Len(nn$) - 1)
l = Len(auxi)
```

```
If l < 2 Then
escapicua = False
Else
c = True
i = 1
k = Int(l / 2)
While i <= k And c
  If Mid(auxi, i, 1) <> Mid(auxi, l - i + 1, 1) Then c = False
  i = i + 1
  Wend
End If
escapicua = c
End Function
```

Os invitamos a usar el algoritmo para averiguar (para números no muy grandes) cuántos pasos son necesarios hasta que un número desemboque en un capicúa sumando de forma reiterada su reverso.

Jugando un poco con este algoritmo se pueden descubrir hechos interesantes.

Llamaré meta al capicúa en el que termina un algoritmo aplicado a un número (semilla) y ruta al conjunto de números que se recorren hasta llegar desde la semilla hasta la meta.

Metas capicúas de dos cifras: Es evidente que los números semilla que desembocan en el mismo capicúa tienen todos la misma suma de cifras y esta es menor que 10. Por ejemplo, 70, 61, 52, 43, 34, 25, 16, 7 desembocan en 77=11*(a+b) con a+b<10

Llegan a 121 los de dos cifras que sumen 10 u 11 y alguno más de tres cifras ¿cuál? Y a 363 los de suma 12 y alguno más de tres cifras, con expresión N=11*(a+b) en el que se cumple a+b>10=11(10m+n)=110m+11n

Metas de tres cifras: Se puede demostrar que sólo son metas los capicúas en los que la cifra del centro es par, como 343, 929, 787,…y por tanto no lo son 232, 878 o 171. Intenta demostrarlo, que no es complicado.

Metas de cuatro cifras: Han de ser múltiplos de 11. ¿Por qué?

Números ilustres: Los números 495 y 1089, están ambos en la misma ruta que desemboca en el 79497. Además, tienen como meta el 1089 los múltiplos de 198 de tres cifras.

Otra curiosidad: Los 10 primeros múltiplos de 1089 llegan todos hasta el 79497.

Si no recuerdas el porqué de que les llame “ilustres” al 495 y al 1089, consulta esta dirección:

http://www.hojamat.es/sindecimales/aritmetica/propuestas/proparit.htm

Ahí te enterarás que 495 es la constante de Kaprekar para tres cifras. Si elegimos como semilla la constante para cuatro cifras 6174, también tiene como meta 79497, lo que nos confirma que ambos algoritmos están relacionados. Es curioso que 6174+4716 = 10890 = 1089+9801.

CONJUNTOS IDÉNTICOS

En algunas cuestiones resulta útil decidir de forma automática si dos conjuntos son idénticos o no. Por ejemplo, en las tablas de multiplicar de los cuerpos finitos, como Z/Z7, es interesante descubrir si

(a) No existen elementos repetidos en ninguna fila o columna
(b) Los elementos de las distintas filas son los mismos.

Si escribimos los dos conjuntos en una hoja de cálculo, en filas paralelas, deberemos comprobar cuatro hechos para decidir si los conjuntos son idénticos o no:

1. No existen elementos repetidos en el primer conjunto
2. Tampoco se repiten los del segundo
3. Todo elemento del primero ha de pertenecer al segundo
4. Todo elemento del segundo ha de pertenecer al primero.

Las cuatro cuestiones las resuelve la función CONTAR.SI. Recorremos todo el primer conjunto y mediante esta función contamos las veces que figuran en el segundo. Si esos valores son mayores que 1, es que existen repetidos en el segundo conjunto, y si es 0, es que falta alguno. Lo deseable, pues, es que todos los contadores presenten el valor 1.

Procedemos de la misma forma, contando las veces que los elementos del segundo conjunto figuran en el primero, y también han de valer 1. Para evitar problemas en las siguientes operaciones que explicaremos, a las celdas vacías también se le debe asignar un 1.

¿Cómo resumimos la situación? Multiplicamos todos los contadores del primer conjunto, y nos ha de resultar la unidad. Ocurrirá lo mismo con el producto de los del segundo, por lo que si multiplicamos ambos productos, obtendremos un criterio para decidir si los dos conjuntos son idénticos: el que el producto final tenga el valor de 1.

ALGORITMOS AYUDADOS

En un comentario publicado en nuestro blog,
Claudio (http://simplementenumeros.blogspot.com/)
nos proponía lo siguiente:

Te envío un problema que mandó Rodolfo Kurchan a la lista de Snark:

Este simpático acertijo me lo envió Michael Reid de EEUU: Colocar los dígitos 0, 1,..., 9, sin repetir en la expresión a^b + c^d + e^f + g^h + i^j para obtener el año actual.

Pensé en aplicar esta idea al 2010. Como estábamos en días navideños, no me apetecía pensarlo mucho. Por otra parte, si intentaba un algoritmo sin preparación, me podía encontrar con 10^10 pasos, lo que era demasiado para cualquier hoja de cálculo.

Así que pensé en ayudar un poco al algoritmo, para ver si entre la máquina y yo lo encontrábamos sin gran esfuerzo. La primera idea fue la de un algoritmo voraz, pero también había que diseñar bastante, y mi cabeza estaba con los villancicos.

Después de cavilar se me ocurrió elaborar una lista de potencias desde 0^0 hasta 9^9 eliminando las mayores de 2010 y casi todas las triviales: 1^3, 1^4,...0^8…La lista estaba compuesta por 1296,1024, 729, 625, 512,…5, 4, 3, 2, 1, 0 (29 en total)

De esta forma, el algoritmo podía traducirse a “Descomponer 2010 en sumandos tomados de una lista”. Aún así, los cálculos tardaban demasiado (29^5 pasos) y los compromisos sociales me esperaban. Lo dejé para otro día.

Reanudada la tarea, impuse la condición de que los sumandos fueran no crecientes, lo que simplificaba la búsqueda, y como 2010/5 = 402 debería haber algún sumando superior a esa cantidad. De esta forma puede seguir recortando pasos.

Al final obtuve una lista de sumandos que comenzaba con

1296	625	81	8	0
1296	625	81	7	1
1296	625	81	6	2

y terminaba con

625	625	625	128	7
625	625	512	243	5
625	625	512	216	32

Ya tenía algo con lo que trabajar. Añadí a cada sumando los dígitos que lo formaban como una potencia:

1296	6 y 4	625	5 y 4	81	9 y 2 3 y 4	8	8 y 1 2 y 3	0
1296	6 y 4	625	5 y 4	81	9 y 2 3 y 4	7	7 y 1	1
1296	6 y 4	625	5 y 4	81	9 y 2 3 y 4	6	6 y 1	2

Ya sólo quedaba elegir las posibilidades en las que no se repitieran los dígitos. Encontré cuatro:

4^5+2^8+9^3+1^7+0^6=2010
4^5+2^8+9^3+1^6+0^7=2010
4^5+2^8+3^6+1^8+0^7=2010
4^5+2^8+3^6+1^7+0^9=2010

De esta forma la hoja de cálculo aportó una base para soslayar mi pereza mental, y yo le ayudé con mi sentido común.

SUMA DE TRES NÚMEROS TRIANGULARES

En 1796, Gauss descubrió que todo entero positivo puede representarse como la suma de tres números triangulares (o menos). Estos pueden ser iguales, y si consideráramos el 0 como triangular, podríamos afirmar que todo número natural es suma exactamente de tres triangulares.

Un ejercicio interesante es el de descubrir un algoritmo que los encuentre. Para hacerlo más formativo podemos basarnos en dos funciones, una para descubrir si un número es triangular y otra que nos devuelve el mayor triangular mayor o igual que un número dado.

Función estriangular

Si un número N es triangular verificará igualdad N=x(x+1)/2, con x y N ambos enteros, lo que obliga a que 8*N+1 sea cuadrado perfecto ¿por qué? Esto nos lleva a este código:

```
Public function estriangular(n) as boolean
dim a
a = Int(sqr(8*n+1))
if a*a=8*n+1 then estriangular = true else estriangular = false
end function
```

Función mayortriang

Para encontrar el mayor triangular contenido en un número N bastará resolver la ecuación N=x(x+1)/2 truncando el resultado a un número entero. Así que quedará:

```
Public Function mayortriang(n)
dim a
a = Int((sqr(8*n+1)-1)/2)
mayortriang=a*(a+1)/2
end function
```

Con esto ya tenemos preparado un algoritmo para OpenOffice.org Calc (fácilmente adaptable a Excel) que encuentre todas las descomposiciones en tres triangulares (incluido el cero):

```
Sub sumatriangulares
Dim i,j,k
dim a,b
i=StarDesktop.CurrentComponent.sheets(0).GetCellByPosition(3,3).value
 (se supone que el número se escribe en la celda D4)
a=mayortriang(i)
for j=0 to a (se recorren los valores posibles del primer sumando)
if estriangular(j) then
b=mayortriang(i-j)
for k=j to b (se recorren los valores posibles del segundo sumando)
if estriangular(k) then
if estriangular(i-j-k) and i-j-k&gt;=k then (el tercer sumando ha de ser triangular)
msgbox(j) (se presenta el resultado)
msgbox(k)
msgbox(i-j-k)
end if
end if
next k
end if
next j
End Sub
```

Pues ánimo y a implementarlo. Puedes añadir una variable que cuente todas las formas de descomposición que tiene un número. Por ejemplo, entre los de tres cifras hay uno que admite 24 sumas distintas de triangulares ¿Cuál?

Otras codificaciones

En PARI

```
isinteger(n)=(n==truncate(n))
isquare(n)= { local(f,m,p=0); if(n==1,p=1,f=factor(n); m=gcd(f[, 2]);
if(isinteger(m/2),p=1));return(p) }
istriang(n)=isquare(8*n+1)
{n=95;for(i=0,n,if(istriang(i),for(j=i,n,if(istriang(j),for(k=j,n,if(istriang(k)&&i+j+k==n,
print(i," ",j," ",k))))))))}
```

En primer lugar se definen ***isinteger***, para ver si un número es entero, ***isquare***, para cuadrado e ***istriang*** para triangular.

Después se recorren ciclos para encontrar tres sumandos triangulares crecientes en sentido amplio. Si se cambia el valor 95 por otro número, dispondremos de un algoritmo que resuelve el problema para ese número.

En WxMAXIMA

```
m:92$
esent(x):=if x=floor(x) then true else false;
escuad(x):=(y:sqrt(x),if esent(y) then true else false);
estriang(x):=if escuad(8*x+1) then true else false;
for i:0 thru m do
(
if estriang(i) then for j:i thru m do
     (if estriang(j) then for k:j thru m do
        (if estriang(k) and i+j+k=m then display(i,j,k)
         )
     )
)$
```

Como en el anterior, se definen las funciones previamente y después se construyen tres bucles. El número se escribe en la primera línea.

Algoritmo a paso de tortuga

La factorización de Fermat siempre se ha presentado como una técnica para representar un número impar como producto de dos de sus factores **sin usar la lista de números primos**. No es el único algoritmo de factorización con esa propiedad. Si extraemos progresivamente el factor más pequeño (mayor que 1) de N aseguraremos que hemos encontrado un número primo sin tener que memorizar la lista de primos

En efecto, la factorización de Fermat no se basa en los factores primos, sino en representar un número impar N como una diferencia de dos cuadrados y después expresar la misma como el producto de una suma por una diferencia, con lo que se logra la factorización:

$N=y^2-x^2=(x+y)(y-x)$, $y>x$

En el caso impar esta operación siempre es posible, porque $N=(N+1)^2/4-(N-1)^2/4$, que da lugar a la factorización N=N.1

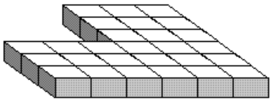
Desde el punto de vista algorítmico, la búsqueda de los valores de x e y adecuados presenta también otra originalidad, y es que se puede organizar de forma bastante eficiente sólo con sumas y comparaciones. En realidad, todos los algoritmos se pueden organizar así, y en caso último mediante la máquina de Turing, pero tampoco hay que llegar tan lejos, y con tantas operaciones se lentifica mucho el proceso. En el caso de la factorización de Fermat se logran resultados bastante aceptables. Es una tortuga, pero algo veloz.

La creación del algoritmo se basa en estos hechos que te pedimos intentes justificar:

(1) El valor de **y**, y por tanto el de **x**, están acotados, y su cota no está excesivamente alejada de n ¿Cuál es y cómo se demuestra?

(2) Para demostrar que un número es cuadrado perfecto, no es necesario dividir ni extraer la raíz cuadrada ¿Cómo se hace?

(3) En realidad, la factorización de Fermat consiste en darle al número impar la figura de escuadra simétrica (gnomon) de varias formas posibles. Observando la imagen puedes resolver las dos primeras cuestiones.



(4) Podemos encontrar la raíz cuadrada entera de un número haciendo crecer de forma simultánea dos sucesiones, una linealmente y otra de forma cuadrática. Esto parece obvio, pero ¿cómo se organiza?

Podemos representar el algoritmo como una carrera, liebre y tortuga, entre $a=x^2$ y $b=y^2$, en la que la tortuga sale con ventaja porque se suma al número N, ya ha de ser $y^2=x^2+N$. Todo esto suena a metáfora, pero funciona.

Daremos a continuación un desarrollo en el que se ocultarán algunos detalles para hacer pensar un poco a quien lo siga.

Obtención de un primer valor de $a=y^2$

Tomamos tres variables, y, a, i

Las iniciamos a

y=1; a=1; i=1

Mientras a no sobrepase a N hacemos crecer estas variables así:

y=y+1; i=i+2; a=a+i

con lo que lograremos el primer cuadrado que es mayor o igual que N

Sólo hemos usado sumas y una comparación. Razona por qué se da este resultado.

Obtención de valores adecuados de $a=y^2$ y de $b=x^2$

Una vez obtenido y^2 iniciamos el crecimiento de la misma forma para x^2

x=1; b=1; j=1

Mientras N+b no sobrepase a y^2, hacemos crecer las variables:

x=x+1; j=j+2; b=b+j

para formar el cuadrado x^2

Si ocurre que N+b=a hemos conseguido una factorización, pues $N=y^2-x^2$

Obtenemos todas las coincidencias

Ya sólo queda hacer crecer **a** y **b** de la misma forma y comprobar si se da la igualdad N+b=a en más ocasiones, y así hasta la cota.

No hemos querido usar el lenguaje algorítmico, y se han ocultado algunos detalles, como qué hacer si N es un cuadrado perfecto. Lo importante de lo explicado es que sólo hemos sumado y comparado. No se ha recurrido a multiplicar, dividir o extraer la raíz cuadrada.

Puedes estudiar más a fondo el algoritmo en el Anexo. Si recorres el código de la macro usada, sólo verás operaciones de sumar. El proceso no es un prodigio de velocidad, pero esto es un divertimiento. La idea de hoy no era correr, sino demostrar una posibilidad. Tampoco corrió Fermat y hay que ver lo que logró.

REPRODUCIR RESULTADOS

Somos muchos en el mundo. Estudiamos en una Facultad de Matemáticas, llevamos años y años enseñándolas, seguimos estudiando distintos temas y leyendo libros de divulgación, entretenimientos o curiosidades, pero nunca hemos publicado un resultado matemático apreciable. Sólo nos queda disfrutar con desarrollos ajenos, resolver placenteramente problemas de más o menos dificultad y… reproducir resultados.

Lo de obtener lo que ya han descubierto otros puede ser formativo y entretenido si lo intentamos con herramientas distintas a las del primero que lo logró. En este blog usamos las hojas de cálculo, lo que nos exige la construcción de tablas en las que se llevan al límite las posibilidades de las funciones que traen implementadas, o bien, que es una tarea más apasionante, implementar algoritmos adecuados mediante macros.

Proponemos una reproducción:

He leído por ahí, en la Wikipedia, Wolfram Mathworld o una página similar, que el número 2011 es estrictamente no palindrómico. Se llaman así los números N que no son palindrómicos (capicúas) para bases comprendidas entre 2 y N-2. No se consideran bases mayores porque todos los números se expresan en ellas como capicúas (se admite que lo son los de una cifra) para bases mayores que N-2. ¿Sabrías razonarlo?

Alguien se ha tomado la molestia de ir probando el 2011, imagino que de forma automática, para todas las bases comprendidas entre 2 y 2010.

¿Puedes reproducir ese resultado con hoja de cálculo?

Para averiguar si un número es estrictamente no palindrómico necesitaremos una función que nos diga si es palindrómico o no en una base dada, y después recorrer todas las bases entre 2 y N-2 para descubrir si hay o no resultados negativos.

Diseñaremos la función ESCAPICUA(n,b), donde n será el número a probar y b la base del sistema de numeración. Esta función nos devolverá un 1 si el número es palindrómico y 0 si no lo es. Usamos 1 y 0 porque son más cómodos que True y False.

Necesitaremos organizar dos fases de cálculo

a) Extracción de las cifras de n en base b y almacenamiento de las mismas en una matriz c

b) Emparejamiento de las cifras de forma simétrica para averiguar si son todas iguales por parejas (caso palindrómico) o bien existe una que no es igual a su simétrica.

Primera fase: extracción de las cifras

Usaremos un algoritmo voraz, en el que **n** va disminuyendo de valor, con lo que la velocidad se acelera. Dividimos en cada paso n entre b, quedando el cociente como nuevo valor de n y el resto como cifra nueva. Cuando el cociente sea cero, paramos.

Puedes estudiarlo en Basic

En el listado hemos copiado n en m para preservar su valor

```
' extraer cifras
nopara=true    Esta variable determina si se para o no el proceso
nc=0           Contador de cifras
while nopara
q=int(m/b):r=m-q*b  Se halla el cociente y el resto de m entre la base
if q=0 then nopara=false  Si el cociente es cero, se para
nc=nc+1:c(nc)=r:m=q   Se incrementa el contador de cifras y se almacena la nueva
wend
```

Segunda fase: Comparación entre cifras

Una vez almacenadas las cifras, si sólo hay una, se declara el número como palindrómico. En caso contrario, si se detecta una desigualdad entre cifras simétricas, se declara como no palindrómico.

En Basic

```
esca=1  Admitimos que es capicúa
if nc>1 then  Si hay más de una cifra, analizamos
for q=1 to int(nc/2)
if c(q)<>c(nc-q+1) then esca=0  En caso de desigualdad, no es capicúa
next q
escapicua=esca
end if
```

Si deseas implementar esta función en tu hoja de cálculo, copia el código completo:

```
Public function escapicua(n,b)
dim c(50)
dim m,q,r,nc,esca
dim nopara as boolean

m=n
' extraer cifras
nopara=true
nc=0
```

```
while nopara
q=int(m/b):r=m-q*b
if q=0 then nopara=false
nc=nc+1:c(nc)=r:m=q
wend
esca=1
if nc>1 then
for q=1 to int(nc/2)
if c(q)<>c(nc-q+1) then esca=0
next q
escapicua=esca
end if
end function
```

Con esta función se puede rellenar una columna que actúe sobre las bases comprendidas entre 2 y N-2. Por ejemplo, en la imagen puedes comprobar que el número 19 es estrictamente no palindrómico:

¿Es N palindrómico?

Escribe el número	19
Base	Es capicúa
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

Los primeros números estrictamente no palindrómicos son:

1, 2, 3, 4, 6, 11, 19, 47, 53, 79, 103… (http://oeis.org/A016038) En esta página de OEIS descubrirás que **son todos primos a partir del 6**. La justificación de esto proviene de que a*b=a(b-1)+a siendo a<b, lo que lo convierte en palindrómico en base b-1 (se expresa como 11). En el caso del cuadrado a*a=(a-1)^2+2(a-1)+1, lo que lo convierte en palindrómico en base a-1, expresado como 121. Luego los compuestos serán paindrómicos en ciertas bases. Puedes leer más detalles en la dirección citada.

Hemos aplicado la prueba a 2011 y, efectivamente, no es palindrómico para ninguna base comprendida entre 2 y 2010.

Con ello hemos reproducido un resultado, con la consiguiente diversión e incremento de nuestra confianza en la comunidad matemática.

Si te atreves, codifica una función ESTRICTCAP, que decida si un número es estrictamente no palindrómico. Bastará programar en Basic lo que en la imagen hemos efectuado con columnas.

Tanto Internet como los libros de divulgación matemática están llenos de listas de números que se caracterizan por ser los únicos que cumplen algún requisito.

La página http://oeis.org/A084687 nos presenta la siguiente lista como la de los números enteros positivos que son múltiplos de los números formados por sus mismas cifras ordenadas en orden creciente:

9513, 81816, 93513, 94143, 95193, 816816, 888216, 933513, 934143, 935193, 941493, 951993, 2491578, 8166816, 8868216, 9333513, 9334143, 9335193, 9341493, 9351993, 9414993, 9519993, 24915798, 49827156, 81666816, 87127446, 88668216, 93333513

Este requisito ha de cumplirse en sentido estricto:

- No pueden contener cifras nulas.
- No pueden poseer ellos mismos las cifras ya ordenadas.

El primer ejemplo de la lista es el número 9513, que no contiene cifras nulas y es múltiplo de 1359, formado por las cifras 9, 5, 1 y 3 ordenadas de forma creciente.

Los cocientes que se forman son “casi todos” iguales a 7. Investiga este hecho si quieres.

Un ejercicio muy formativo es el de obtener esa misma lista con nuestros propios instrumentos, que aquí será la hoja de cálculo. Para ello debemos organizar muy bien el proceso, y en esta tarea aprenderemos de Matemáticas y de programación mucho más de lo que nos creemos.

Presentamos una organización del proceso de obtención de la lista presentada, aunque sería deseable que nuestros lectores no siguieran leyendo y pasaran a su propia organización. Así también ellos, como nosotros, aprenderían probando.

Un posible esquema sería el siguiente:

Obtención de la lista de números

- *Se recorren todos los números A desde un inicio hasta un número final.*
- *Para cada uno se realizan estas operaciones:*
- *Calcular el número de cifras de A*
- *Extraer todas las cifras de A. Si alguna es cero se rechaza el número.*
- *Ordenar las cifras*
- *Formar con esas cifras un nuevo número B*
- *Si A=B se rechaza el número.*
- *Si A es múltiplo de B se incorpora A a la lista.*
- *Se pasa al siguiente número*

Si te interesa la programación en Basic, puedes estudiar el siguiente código comentado para OpenOffice.org Calc:

Funciones auxiliares

Para saber si m es múltiplo de n. Devuelve 1 si lo es, y 0 si no lo es

```
Public function esmultiplo(m,n)
if m=int(m/n)*n then esmultiplo=1 else esmultiplo=0
end function
```

Para contar el número de cifras

```
Public function numcifras(n)
numcifras=int(log(n)/log(10))+1
end function
```

Extrae la cifra de orden n de un número m

```
Public function cifra(m,n)
dim a,b
a=10^(n-1)
b=int(m/a)-10*int(m/a/10)
cifra=b
end function
```

Algoritmo de búsqueda

Sub busquedas
dim n,m,i,j,k,l,a,b,fila,p,q
dim ci(12)

Lee el inicio (celda G7) y el final (celda H7)

n=StarDesktop.CurrentComponent.sheets(0).GetCellByPosition(6,6).value
m=StarDesktop.CurrentComponent.sheets(0).GetCellByPosition(7,6).value
fila=8

Recorre los numeros

for i=n to m

Extrae cifras y las ordena

Extrae cifras

k=numcifras(i)
for l=1 to k
ci(l)=cifra(i,l):if ci(l)=0 then exit sub   'no se admiten cifras nulas
next l

Las ordena

if k>=1 then

for j=1 to k-1
for p=2 to k
if ci(p-1)<ci(p) then b=ci(p-1):ci(p-1)=ci(p):ci(p)=b
next p
next j

end if

Construye el número con cifras ordenadas

q=0
for j=1 to k
q=q+ci(j)*10^(j-1)
next j

‘si es múltiplo, lo presenta en columna

```
if esmultiplo(i,q)=1 and i<>q then
StarDesktop.CurrentComponent.sheets(0).GetCellByPosition(6,fila).value=i
StarDesktop.CurrentComponent.sheets(0).GetCellByPosition(7,fila).value=q
fila=fila+1
end if

next i

end sub
```

Ánimo y a estudiarlo, que contiene bastante información valiosa.

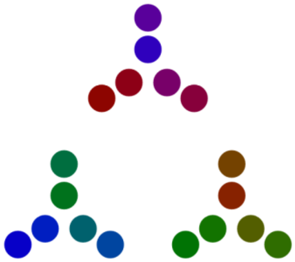
UNA HUMILDE IMITACIÓN

Desde hace unas semanas circula por la red un atractivo vídeo en el que se visualiza la factorización de los números como conjuntos de puntos en forma de árboles poligonales circulares muy cercanos a los conjuntos fractales

http://miscuriosidadesmatematicas.blogspot.com.es/2012/11/diagrama-animado-sobre-factorizacion-de.html

http://www.datapointed.net/visualizations/math/factorization/animated-diagrams/

http://mathlesstraveled.com/2012/10/05/factorization-diagrams/



En la imagen se puede ver el desarrollo para el número 18, en el que los niveles están definidos por los factores 3, 3 y 2.

Como en este blog no nos salimos de los números y de la hoja de cálculo nos planteamos un reto:

¿Hasta dónde podíamos imitar estos esquemas usando tan sólo la programación de una hoja de cálculo?

Es evidente que el resultado obtenido ha de ser muy inferior al original, y que el interés de esta tarea está en la adaptación a una herramienta menos potente. Por tanto, quien no esté interesado en esta programación es mejor que disfrute del vídeo original sin embarcarse en el proceso que aquí hemos seguido.

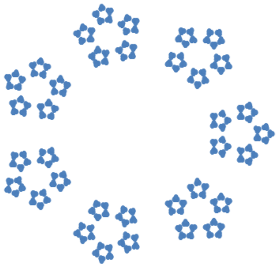
El resultado lo tienes en http://hojamat.es/blog/arbofactor.xlsm

Ideas previas

Para imitar lejanamente el vídeo necesitaremos concretar aspectos del problema en sí mismo (representar cada factor como un polígono) y después superar las carencias de la hoja de cálculo (en esta ocasión, dado el distinto comportamiento de las mismas en los gráficos, lo hemos desarrollado sólo en Excel)

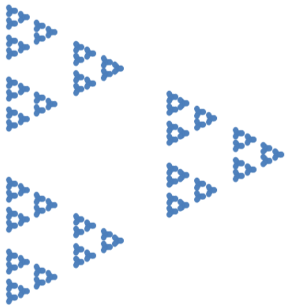
El gráfico

La única posibilidad que nos ofrece Excel para estas representaciones es el gráfico de dispersión. Tiene la ventaja de no depender del orden de los datos y adaptarse muy bien al uso de coordenadas cartesianas (X,Y). También permite dejar la zona en blanco y ocultar los ejes. Por contra, presenta gran dificultad en cambiar el tamaño de los distintos puntos según el número de factores. Todos los esquemas, pues, presentarán el mismo tamaño en los puntos



Aquí puedes ver el esquema para 1050. Se puede observar que los últimos triángulos de puntos parecen confundirse, por no haber controlado el tamaño.

Salvo este inconveniente, las figuras resultan atractivas. Las hemos orientado circularmente en lugar de buscar siempre la vertical. En la siguiente imagen puedes observar la estructura casi fractal que presentan los números que como el 486 contienen potencias altas de primos.



Lo único que hay que explicar del gráfico es que su área de datos está formada por las columnas B y C en las que volcaremos las coordenadas adecuadas.

El algoritmo

Sacar los primos

Necesitamos, en primer lugar, la lista de factores primos del número, con repetición y en orden decreciente. Hemos aludido bastante en este blog a estas técnicas, por lo que a nuestros seguidores les resultará familiar. Normalmente se comienzan a buscar los factores pequeños, pero en este trabajo, por razones estéticas, se comienza por los mayores. Por eso al final de la rutina se invierte el orden.

```
Sub sacaprimos(n)
Dim f, a, indi

a = n
f = 2: nprimos = 0
indi = 0

While f * f <= a
While a / f = Int(a / f)
  indi = indi + 1
 ff(indi) = f  'La variable ff va recogiendo los primos
a = a / f
Wend

If f = 2 Then f = 3 Else f = f + 2
Wend
If a > 1 Then  'Último factor
indi = indi + 1
ff(indi) = a
```

End If

' se ordenan al revés

```
nprimos = indi
For indi = 1 To nprimos
xx(indi) = ff(nprimos - indi + 1)
Next indi
For indi = 1 To nprimos
ff(indi) = xx(indi)
Next indi

End Sub
```

Después de esta rutina tendremos el número de factores primos en la variable **nprimos** y sus valores en **ff(i)**. En este caso no nos interesan los exponentes.

Recorrido en cada factor

Una vez obtenidos los factores primos necesitamos convertirlos en polígonos. Para cada uno harán falta las coordenadas del centro (xx(i),yy(i)), su radio rr(i) y un contador de puntos ii(i) que nos evite el problema de los decimales al final de cada polígono. En cada paso incrementaremos el ángulo de giro necesario para que se forme el polígono y el contador de lados

En la imagen hemos comenzado con ff(1)=13, por lo que los elementos se incrementarán así:

```
ii(i) = ii(i) + 1
aa(i) = aa(i) + dospi / 13
```

Marcado el ángulo que va girando el punto pasamos de coordenadas polares a cartesianas y volcamos el punto en la columnas B y C, donde lo capturará el gráfico y aparecerá un punto nuevo.

```
fila = fila + 1
px = xx(i) + rr(i) * Cos(aa(i))
py = yy(i) + rr(i) * Sin(aa(i))
```

```
ActiveWorkbook.Sheets(1).Cells(fila, 2).Value = px
ActiveWorkbook.Sheets(1).Cells(fila, 3).Value = py
```

Esto se repite tantas veces como indique el factor y se formará el polígono básico

Algoritmo de cambio de nivel

Aquí reside el nudo de esta programación. Es un caso claro de procedimiento recursivo, pero como hay que gestionar bastantes parámetros lo hemos dejado para una posible extensión y se ha usado en su lugar un esquema muy sencillo que ya se ha publicado en este blog: la subida y bajada de nivel.

Procedimientos iniciales: definir primer radio, sacar los primos…

Mientras haya un nivel activo (nivel>0)
Incrementamos el ángulo y el contador para avanzar en el polígono
Si se llega al final del polígono
Hemos terminado y bajamos de nivel (nivel=nivel-1)

En caso contrario pueden ocurrir dos cosas:

Si hemos llegado al último factor hay que imprimir el punto volcándolo en las columnas B y C

Si no hemos llegado hay que subir de nivel(nivel=nivel+1)

Esto significa que hay que determinar nuevos centro y radio

Fin del Mientras

Fin de la rutina

No incluimos el código y preferimos que las personas interesadas lo estudien en el archivo de Excel.

Animación

Para conseguir la animación basta con una estructura repetitiva tipo FOR-NEXT y la creación de pausas para conseguir el efecto de transición continua. El problema radica en que cualquier operación aparentemente sencilla puede borrar el gráfico y perderse su persistencia en nuestra retina. Hemos tenido que acudir al recálculo para refrescarlo y que no desaparezca.

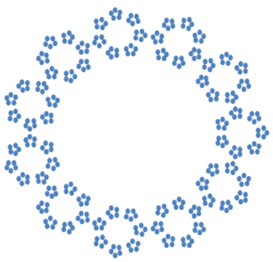
La pausa se consigue leyendo el reloj e introduciendo a la hoja en un bucle continuo hasta que transcurra la pausa. Queda así:

```
Call arbol  'se construye el árbol de factores
t1 = Timer ' leemos el reloj y tomamos nota en t1 y t2
t2 = t1
While t2 - t1 < pausa: t2 = Timer: Calculate: Wend  'bucle continuo de recálculo
Call borrar 'terminada la pausa se borra el gráfico
```

Controles

Nos gusta tener todo el poder posible sobre la hoja de cálculo. Por eso se han añadido los controles de **Reducción**, que se puede cambiar (no en la animación) para mejorar la estética del gráfico y **Pausa**, que ralentiza o acelera la animación.

A quienes hayan llegado hasta aquí les recomendamos el estudio de los detalles del código y les invitamos a intentar mejorarlo.



ALGORITMO DE EUCLIDES BINARIO

En este blog nos motiva mucho el construir un esquema en hoja de cálculo que explique de la mejor forma posible el funcionamiento de un proceso o un algoritmo. Lo haremos hoy con la variante binaria del Algoritmo de Euclides.

Restar en lugar de dividir

Ya se sabe que el algoritmo de Euclides por divisiones se puede sustituir por otro efectuado a base de restar los dos números. Parece más lento, pero al ahorrar divisiones puede resultar más eficiente. Hace sesenta años lo usábamos los alumnos de Bachillerato (con once añitos) para comprobar si dos segmentos eran inconmensurables o no. Llevábamos uno sobre otro y nos quedábamos con la diferencia. Si al reiterar llegábamos al segmento nulo, es que tenían una medida común. Aquello era Geometría de Euclides pura.

Si el algoritmo clásico se organizaba a partir de la división euclídea, en esta variante se usa la resta. Así, para encontrar MCD(96,36) por divisiones sería: 96=2*36+24; 36=1*24+12; 24=2*12+0, luego el MCD(96,36)=12. Por restas formaríamos estas parejas: (96,36), (60,36), (36,24), (24,12), (12,12), (12,0), llegando al mismo resultado.

Eliminar el factor 2 siempre que se pueda.

Ya sabemos que en computación con sistema de numeración binario la división y la multiplicación por 2 se reducen a trasladar un lugar a la derecha o izquierda del dígito que se multiplica. Por eso, es interesante dar protagonismo al número 2 en los cálculos. Una primera idea es que si expresamos los dos números A y B de la forma $A=2^m*p$, $B=2^n*q$, con p y q los mayores divisores impares, el M.C.D se puede encontrar con dos cálculos por separado. Por una parte quedándonos con el menor exponente del 2 y por otra hallando MCD(p,q).

En el algoritmo que estamos presentando, se elimina en primer lugar la potencia de 2 común a ambos números, y se toma nota de ella. Una vez eliminada, el factor 2 no va a influir en el resultado, y cuando aparezca en alguno de los dos números se podrá igualmente suprimir. En términos binarios suprimir un 2 es trasladar los dígitos una posición hacia la derecha.

En Wikipedia y otras páginas que hemos consultado expresan lo anterior en forma de tres reglas.

http://en.wikipedia.org/wiki/Binary_GCD_algorithm. No son difíciles de razonar.

(1) Si A y B son ambos pares se cumple MCD(A,B)=2*MCD(A/2,B/2)

Esto justifica que el primer paso que demos sea el de separar las potencias de 2.

(2) Si A es par y B impar se tiene MCD(A;B)=MCD(A/2,B)

Igualmente se aplicaría si B es par y A impar. En virtud de esta regla eliminaremos todos los factores 2 una vez que se ha separado la potencia común.

(3) Si ambos A y B son impares y A<B, MCD(A,B)=MCD(B-A,A). Si es B<A, MCD(A,B)=MCD(A-B,B)

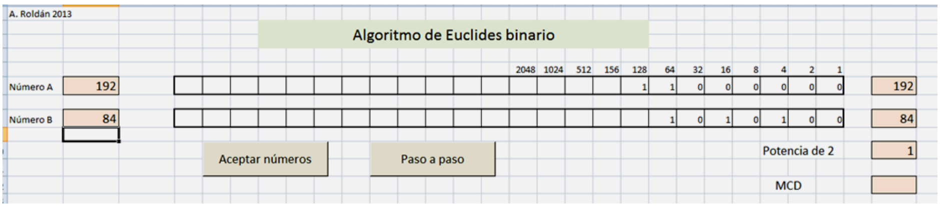
Esta es la esencia del algoritmo de Euclides, restar ambos números.

Eliminar la potencia de 2 común (Regla 1)

Hemos creado una demo en hoja de cálculo para seguir visualmente los pasos del algoritmo binario. Lo puedes descargar desde la dirección

http://hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#euclibin

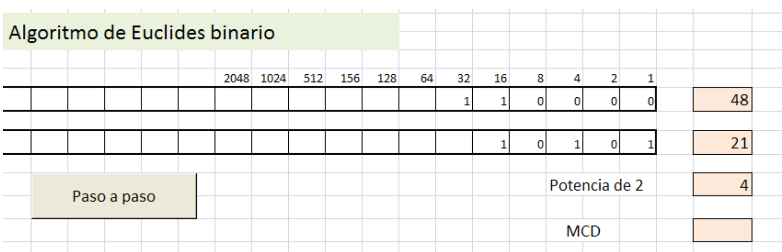
Como nuestro objetivo es visual, desarrollaremos ahora los pasos sugeridos pero en forma de esquema. En una hoja de cálculo se escribirán los dos números y con un botón iterativo se irán avanzando pasos hasta llegar al MCD. Esta primera fase de eliminar el factor común lo puedes ver en las imágenes siguientes:



En primer lugar hemos escrito los números 192 y 84. Al pulsar sobre el botón ***Aceptar números*** se han convertido ambos en binario y se han reconstruido a la derecha. La potencia de 2 común figura al principio con el valor 1.

Observamos que ambos números terminan en dos ceros (en binario), luego compartirán el factor 2^2=4.

Según estamos explicando, el primer paso del algoritmo binario es eliminar el factor 2 común. Si ahora usamos el botón ***Paso a paso*** dos veces veremos que los dígitos de ambos números se mueven dos posiciones a la derecha y que la potencia de 2 se convierte en 4.



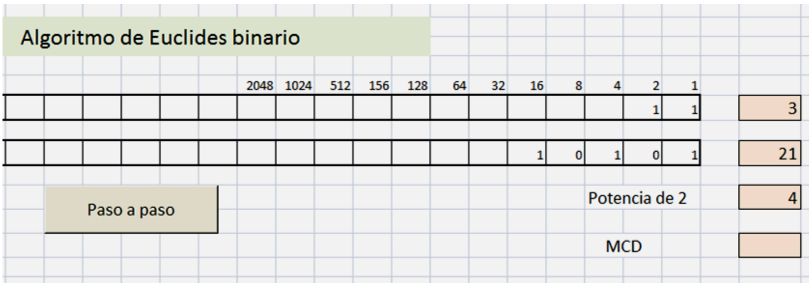
A la derecha figuran los números ya simplificados, 48 y 21, y la potencia de 2, que nos servirá al final.

Resto de pasos

Ahora la estrategia es triple:

(1) Si el primer número contiene aún potencias de 2, se eliminan (Regla 2)

Observa que en nuestro ejemplo el número 48 termina en cuatro ceros, luego al cabo de cuatro toques del botón ***Paso a paso*** desaparecerán.



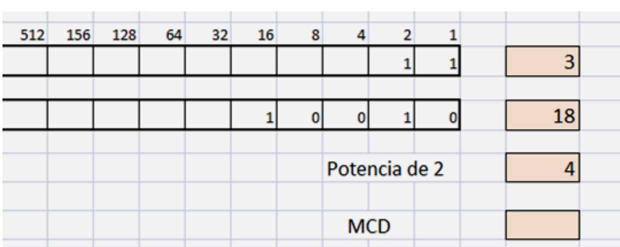
(2) Se opera igual con el segundo: se le elimina el factor 2 (Regla 2)

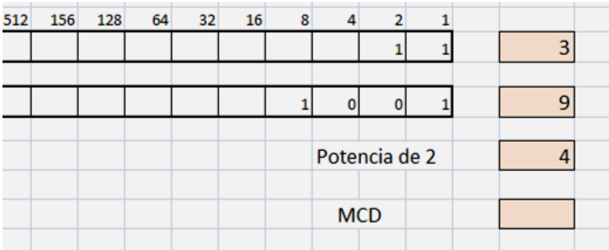
En nuestro ejemplo ya no quedan factores 2 (por ahora)

(3) Si ambos son impares, se sustituye el mayor de ellos por la diferencia entre ambos.

Este es el núcleo del algoritmo de Euclides por diferencias. En la práctica quizás haya que intercambiar los valores de A y B, pero no entraremos en esos detalles.

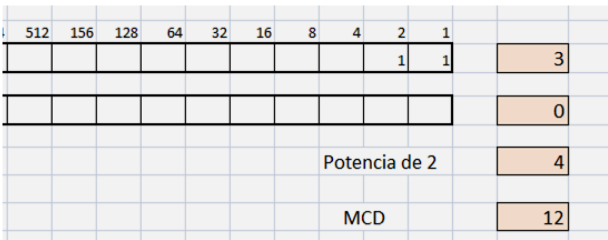
Al restar dos impares se **producirán nuevos factores 2**, por lo que en la siguiente pasada del algoritmo los eliminará. Lo ves en las siguientes dos imágenes:





Reiteramos estas tres reglas hasta que lleguemos al valor 0. En ese momento el algoritmo construye el M.C.D. multiplicando el resultado por la potencia de 2 que tenía almacenada.

Aquí lo tienes:



Visto así, en hoja de cálculo, no parece ser nada del otro mundo, pero todas las operaciones que realiza son altamente eficientes en el sistema de numeración binario. Por algo lo introdujo el programador israelí Stein en 1967. Aquí sólo se nos queda como un tema de cultura matemática, pero es divertido implementarlo.

Versión recursiva

Lo que hemos desarrollado, con un botón que en cada paso reacciona según lo que se encuentra, nos permite sospechar que todo esto se resuelve también mediante una función recursiva. Es cierto, y está publicado. Lo que haremos aquí es adaptarlo al Basic de Excel y Calc:

```
Public Function mcdbin(m, n)
Dim mb

'Si son iguales, hemos llegado al MCD
If m = n Then
mb = m

'Si ambos son divisibles entre 2, se saca ese factor
ElseIf m / 2 = m \ 2 And n / 2 = n \ 2 Then
mb = 2 * mcdbin(m / 2, n / 2)
```

```
'El primero contiene un 2. Se elimina
ElseIf m / 2 = m \ 2 Then
mb = mcdbin(m / 2, n)

'Operamos de igual forma con el segundo
ElseIf n / 2 = n \ 2 Then
mb = mcdbin(m, n / 2)

'Ambos son impares. Se restan
Else
If m > n Then mb = mcdbin(m - n, n)
If n > m Then mb = mcdbin(m, n - m)
End If

'La función recoge el valor de mb
mcdbin = mb
End Function
```

Tiene toda la elegancia de las funciones recursivas

(ver http://hojaynumeros.blogspot.com.es/2012/03/funciones-recursivas-en-las-hojas-de.html)

En este caso resulta un poco complicada, pero funciona muy bien. Si te apetece, estudia la versión clásica y recursiva:

```
Public Function mcdeuclid(m, n)
Dim mb

'Si uno es múltiplo de otro, obtenemos el MCD
If m / n = m \ n Then
mb = n
ElseIf n / m = n \ m Then
mb = m
Else
'En caso contrario, se restan
If m > n Then mb = mcdeuclid(m - n, n)
If n > m Then mb = mcdeuclid(m, n - m)
End If
'La función recoge el valor de mb
mcdeuclid = mb
End Function
```

Ambas están implementadas en la herramienta que ofrecemos.

	Versión recursiva	
Variante binaria	Función MCDBIN	22
Variante clásica	Función MCDEUCLID	22

LAS OLVIDADAS FRACCIONES CONTINUAS

PRESENTACIÓN

¿Sabes qué significa este desarrollo y cómo obtenerlo?

$$\frac{1280}{345} = 3 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{4 + \cfrac{1}{2}}}}}$$

Si no conoces la teoría inténtalo según tus conocimientos.

Puedes comenzar así:

1280/345 = 3 + 245/345 = 3 + 1/(345/245) =3 + 1/(1+100/245) =

3 + 1/(1 + 1/(245/100)) =…

DESARROLLO

Llamamos fracción continua a la expresada de esta forma:

$$a + \cfrac{1}{b + \cfrac{1}{c + \cfrac{1}{d\ldots}}}$$

donde **a** es entero y **b, c**…son enteros positivos llamados **cocientes**. Toda fracción ordinaria se puede expresar de esta forma, y todo número irracional admite aproximaciones mediante desarrollos de este tipo. Las fracciones continuas se usan cuando se desea manejar un representación de los números reales independiente del sistema de numeración (salvo en la expresión de los cocientes).

No es este libro el espacio más adecuado para estudiar todo su desarrollo teórico. Nuestro interés aquí será la implementación de los algoritmos necesarios en hoja de cálculo para desarrollar un número en fracciones continuas y las aplicaciones que derivan de ello.

Dos líneas podemos seguir en la obtención de los cocientes. Una está esbozada en la página anterior y la desarrollaremos más adelante. La otra se basa en el algoritmo de Euclides.

Método del algoritmo de Euclides

Si consultas la teoría descubrirás que los cocientes a, b, c,… son los que aparecen en el algoritmo de Euclides para el cálculo del m.c.d. de dos números. Así, por ejemplo, para encontrar el m.c.d. de 345 y 1280 en el algoritmo se obtienen los siguientes cocientes:

	3	1	2	2	4	2	0	0	0
1280	345	245	100	45	10	5	0	0	0
245	100	45	10	5	0	0	0	0	0

En el desarrollo mediante fracciones continuas de 1280/345 vuelven a aparecer los mismos cocientes 3, 1, 2, 2,… ¡porque se trata del mismo algoritmo orientado de forma diferente! En la siguiente imagen, capturada de una hoja de cálculo

Numerador	1280		Fracción continua:	3	1	2	2	4	2
				3	4	11	26	115	256
Denominador	345			1	1	3	7	31	69

puedes comprobar la evidente igualdad de la serie de cocientes. Comprueba que, efectivamente, es válido este desarrollo:

$$\frac{1280}{345} = 3 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{4 + \cfrac{1}{2}}}}}$$

El algoritmo de Euclides lo puedes encontrar implementado en varios sitios, por ejemplo, en nuestra página *Hojamat.es*

http://hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm

Como el algoritmo de Euclides tiene siempre un final en un número finito de pasos, las fracciones continuas de cualquier número racional poseerán también un número finito de cocientes.

Método de la parte entera

En la primera página de este tema comenzamos a desarrollar otro método: consiste en ir encontrando la parte entera de las fracciones e invirtiendo el resto R = 1/1/R. Este método es recomendable para desarrollar un número expresado en el sistema decimal, pero con las hojas de cálculo **no es exacto**, pues se van acumulando errores de redondeo y truncamiento.

¿Cómo se pueden organizar los datos?

Escribe aquí el número deseado								
3,71014493		1,41	2,45	2,22	4,5	2	0	0
	Fracción continua:	3	1	2	2	4	2	0

En la imagen vemos que en la fila inferior se van situando las partes enteras del número propuesto y de los que van apareciendo en la fila superior. Estos números, que en la imagen son 3, 1, 2,2,…son los cocientes del desarrollo.

Los números de la fila superior son los inversos de los restos que se producen al restar las partes enteras. No es difícil de organizar en una hoja de cálculo, porque basta ir copiando las dos fórmulas de parte entera del cociente e inverso del resto para conseguir el desarrollo, sujeto, como hemos dicho a errores de truncamiento y redondeo.

REDUCIDAS

Para construir una pieza, un tornero ha de ajustar unos engranajes de forma que mientras uno gire 2009 vueltas, el otro sólo recorra 2000. En este caprichoso encargo, los números son primos entre sí, por lo que no se pueden simplificar, y el tornero carece de engranajes de 2000 ó de 2009 dientes.

Le pide consejo al oficial. Éste hace unos cálculos y le ofrece la solución: "Usa un engranaje de 222 dientes y otro de 223, que nadie lo va a notar"

¿Qué operación hizo el oficial? ¿Por qué estaba seguro de que la pieza saldría bien fabricada?

Hemos desarrollado 2009/2000 en forma de fracción continua, formada por los cocientes [1, 222, 4,2]

Puedes reproducir los resultados con las hojas de cálculo **fraccont.xls** y **fraccont.ods** contenidas en la dirección

http://hojamat.es/sindecimales/aritmetica/herramientas/herrarit.htm

Debajo de los cocientes aparecen una serie de fracciones, llamadas **reducidas o convergentes**, 1/1, 223/222, 893/889, que se van aproximando a 2009/200. Entre ellas figura 223/222, la solución propuesta por el oficial. Puedes ver esta aproximación en los desarrollos decimales que figuran debajo.

Estas reducidas se forman calculando fracciones parciales de izquierda a derecha:

1=1;   1+1/222=223/222;   1+1/(222+1/4) = 1+1/(889/4) =

1+4/889 = 893/889

La hoja de cálculo **fraccont.ods** (en su hoja dedicada a números fraccionarios) logra estas reducidas mediante un algoritmo clásico llamado de “los cumulantes”. Consiste en construir dos sucesiones recurrentes del tipo

$P_n = P_{n-1}*a_n+p_{n-2}$

siendo a_n la sucesión de cocientes de la fracción continua, precedidos en la primera fila por 0 y 1 y en la segunda por 1 y 0. Como ejemplo, si se aplican los cumulantes a la sucesión 1, 1, 1, 1,1…. Resulta la sucesión de Fibonacci 1, 1, 2, 3, 5,8…

Puedes seguir estos cumulantes en las filas que contienen los numeradores y denominadores de las reducidas en el caso de la fracción que usamos al principio, 1280/345. Cada número de las filas de reducidas se ha formado multiplicando el anterior por el cociente de la primera fila y sumando el penúltimo elemento

		3	1	2	2	4	2
0	1	3	4	11	26	115	256
1	0	1	1	3	7	31	69

11 = 2*4+2, 115=26*4+11, 31=7*4+3, y así con todos. No es difícil organizar esto con hoja de cálculo.

Las reducidas permiten la aproximación a una fracción con numerador y denominador grandes mediante otras que están construidas con números más pequeños. Esta utilidad la usaban los torneros cuando carecían de ruedas de determinado número de dientes y debían sustituirlas, con un pequeño error, por otras ruedas más pequeñas.

Por ejemplo, 2009/2000 se puede sustituir por 223/222 con un error inferior a 0,000005.

La reducidas son alternativamente mayores y menores que la fracción dada, y se acercan a ella, pues la diferencia entre dos reducidas es siempre igual a la unidad dividida entre el producto de sus denominadores.

ECUACIONES DIOFÁNTICAS

¿Cómo se pueden repartir 5957 objetos en lotes de 161 y de 182 objetos respectivamente, sin que sobre ni falte ninguno?

Puedes usar la fuerza bruta de las hojas de cálculo (tabla de doble entrada, multiplicaciones y sumas hasta ver el total 5957).

También dispones de la herramienta Solver. Aquí tienes una imagen de su planteamiento:

Con estas herramientas obtendrías las soluciones X=11 Y=23

¿Cómo lo resolverías sin ordenador?

Otra aplicación importante de las fracciones continuas y sus reducidas es la de resolver ecuaciones diofánticas lineales del tipo Ax+By=C, en las que C es múltiplo del MCD de A y B (que son las únicas que poseen solución). Quiere esto decir que A, B y C se pueden simplificar hasta conseguir que MCD(A,B)=1. En lo que sigue supondremos que esto se cumple.

Efectivamente, en un apartado anterior se vio que la diferencia entre dos reducidas consecutivas equivalía a una fracción de numerador la unidad y de denominador el producto de sus denominadores. Esta propiedad también se cumple entre la última reducida y la fracción dada.

Vemos cómo se aprovecha esta propiedad para resolver la ecuación.

Sea, por ejemplo, la ecuación 244X+108Y=112.

Simplificamos: 61X+27Y=28, con MCD(61,27)=1

Fracción continua:				2	3	1	6
	Algoritmo de Euclides		61	27	7	6	1
			7	6	1	0	0
	Reducidas	0	1	2	7	9	61
		1	0	1	3	4	27

Buscamos las reducidas de la fracción 61/27 y elegimos la última 9/4

Y se cumplirá, según la propiedad citada, que 61*4-27*9=1, luego 4 y -9 serán las soluciones de 61X+27Y=1. Bastará multiplicar por el término independiente 28 para obtener una solución: X=4*28 = 112 e Y=-9*28 = -252

Las demás soluciones se obtienen mediante las paramétricas.

X=112-27t

Y=-252+61t

Si se desean soluciones positivas deberemos ajustar el parámetro t

En el ejemplo propuesto en la entrada anterior hay que resolver la ecuación diofántica 182X+161Y= 5957. Si usas la hoja de cálculo **diofant1.ods** (OpenOffice) o la **diofant1.xls** (Excel) contenidas en la dirección

http://www.hojamat.es/sindecimales/aritmetica/herramientas/herrarit.htm

verás que las soluciones dadas en primer lugar son X=6808 Y=-7659. Para conseguir dos soluciones positivas hay que jugar bastante con el parámetro T. Intenta conseguirlo, que no es inmediato. Una solución sería X=23 Y=11.

APROXIMACIÓN DIOFÁNTICA

*¿Sabías que la fracción **3650401/2107560** es una muy buena aproximación de la raíz cuadrada de 3 (coinciden en los trece primeros decimales)? ¿Cómo se ha obtenido?*

Cualquier número expresado en forma decimal puede representarse mediante una fracción continua. Si el número es racional, ésta será finita, pero si es irracional no podrá serlo, y tendríamos que prolongar el desarrollo de la fracción continua hasta el infinito. En los siguientes párrafos veremos cómo.

Un caso muy interesante es el de los irracionales cuadráticos, que, como demostró Lagrange, presentan desarrollos periódicos.

¿Cómo desarrollar un decimal cualquiera en fracción continua exacta (caso racional) o aproximada (si es irracional)?

La idea es: separamos la parte entera y la parte decimal del número N=e+d, y la decimal la expresamos así: N=e+1/(1/d). Volvemos a separar parte entera y decimal de 1/d y reiteramos, con lo que irán apareciendo los cocientes enteros de una fracción continua.

Probamos con la raíz cuadrada de 3. Los cálculos serían:

1,73205080757 = 1+ (1/(1/1,73205080757) =

1+ 1/1,36602540378 = 1+ 1/(1+1/2,73205080760) =

1+1/(1+1/(2+1/1,36602540378) =…

Al salir este último número se descubre la periodicidad, luego 1,73205080757 equivale a la fracción continua

[1, 1, 2, 1, 2, 1,2,….] que es periódica por tratarse de un irracional cuadrático.

Para evitar los errores de truncamiento y redondeo deberás organizar los cálculos sin acudir a su expresión decimal, manteniendo el radical cuadrático en todos ellos. En este caso habrá que acudir en cada paso a la racionalización de los denominadores mediante el producto por el radical conjugado.

Incluimos a continuación el desarrollo para √8, en el que se van destacando los cocientes obtenidos:

$\sqrt{8} = 2 + \frac{1}{1/\left(\sqrt{8}-2\right)} = 2 + \frac{1}{\left(\sqrt{8}+2\right)/4}$ luego $a_1 = 2$

$$\frac{\sqrt{8}+2}{4} = 1 + \frac{1}{1/\left(\left(\sqrt{8}-2\right)/4\right)} = 1 + \frac{1}{\sqrt{8}+2}$$ luego $a_2=1$

$$\frac{\sqrt{8}+2}{1} = 4 + \frac{1}{1/\left(\sqrt{8}-2\right)} = 4 + \frac{1}{\left(\sqrt{8}+2\right)/4}$$ luego $a_3 = 4$

Se observa que llegamos a la misma expresión que cuando obtuvimos $a_1 = 2$, luego hemos llegado a la periodicidad, y

$\sqrt{8} = [2, 1, 4, 1, 4,\ldots]$

A continuación destacamos algunos desarrollos importantes:

$$\sqrt{2} = [1,2,2,2,2\ldots]$$

Números metálicos

Número de oro

$$\phi = \frac{\sqrt{5}+1}{2} = [1,1,1,1,1\ldots]$$

Número de plata

$$\varphi = 1+\sqrt{2} = [2,2,2,2\ldots]$$

Número de bronce

$$\psi = \frac{\sqrt{13}+3}{2} = [3,3,3,3\ldots]$$

Las hojas de cálculo

http://www.hojamat.es/sindecimales/aritmetica/herramientas/hoja/fraccont.xls

http://www.hojamat.es/sindecimales/aritmetica/herramientas/hoja/fraccont.ods

automatizan el proceso. Si las descargas recuerda que están compuestas de dos hojas, una para números fraccionarios y otra para racionales. Si en la primera escribes =RAIZ(3) obtendrás las aproximaciones (leyendo las reducidas) a la raíz cuadrada de 3.

El carácter aproximado de los cálculos produce que se rompan las posibles periodicidades después de muchos pasos.

ECUACIÓN DE PELL

Se llama ecuación de Pell (por error, porque Pell no la estudió) a la ecuación diofántica cuadrática $X^2 - DY^2 = 1$, con X e Y variables enteras y D número entero positivo no cuadrado perfecto. Existe una variante con el segundo miembro -1 que se resuelve de forma similar, con algunas restricciones, y también se consideran los casos en los que se trate de cualquier número entero.

En su resolución hay que distinguir dos problemas:

Primera solución

Una primera solución no es difícil de encontrar en general.

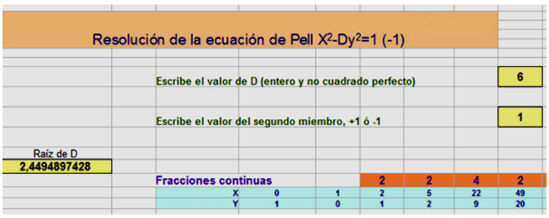
(a) Puedes acudir a un simple tanteo entre cuadrados perfectos. Por ejemplo, una solución de $X^2 - 6Y^2 = 1$ es $X_0=5$ $Y_0=2$. Con una hoja de cálculo no es tarea muy complicada.

(b) Las fracciones continuas también son útiles en la resolución de esta ecuación. Basta para ello desarrollar la raíz cuadrada de D mediante ellas y, según vimos en una entrada anterior, aprovechar la periodicidad del desarrollo. En el caso de la ecuación de Pell basta tomar las reducidas anteriores a la finalización del primer periodo.

2	2	4	2	4	2	4	2
2	5	22	49	218	485	2158	4801
1	2	9	20	89	198	881	1960
-2	1	-2	1	-2	1	-2	1
	Solución		Solución		Solución		Solución

En la imagen observarás que la solución $X_0=5, Y_0=2$ aparece antes del final del primer periodo [2,4] en el desarrollo por fracciones continuas. Después siguen otras: X=49, Y=20, X=485, Y=198, etc.

En nuestro modelo de hoja de cálculo que recomendamos más abajo basta escribir el valor de D y el segundo miembro +1 ó -1 y la hoja se encarga de desarrollar la raíz cuadrada de D mediante fracciones continuas:


Resolución de la ecuación de Pell $X^2-Dy^2=1$ (-1)

Escribe el valor de D (entero y no cuadrado perfecto) 6

Escribe el valor del segundo miembro, +1 ó -1 1

Raíz de D
2,4494897428

Fracciones continuas			2	2	4	2
X	0	1	2	5	22	49
Y	1	0	1	2	9	20

Siguientes soluciones

Según la teoría del anillo Q(√D), que no podemos desarrollar aquí, las primeras soluciones, escritas como $X_0+Y_0\sqrt{D}$ constituyen una unidad del anillo, y también lo serán todas sus potencias, por lo que las siguientes soluciones provendrán de los desarrollos de las expresiones

$$(X_0+Y_0\sqrt{D})^n$$

agrupando después los términos que no contienen el radical como valor de Y y los que sí lo contienen como valor de X. Este método puede ser fatigoso, por lo que es mejor ir obteniendo las distintas soluciones por recurrencia. En efecto, de la anterior consideración se deduce que

$$X_n+Y_n\sqrt{D} = (X_{n-1}+Y_{n-1}\sqrt{D})\,(X_0+Y_0\sqrt{D})$$

O bien, separando términos:

$$X_n = X_{n-1}X_0+Y_{n-1}Y_0D$$

$$Y_n = X_{n-1}Y_0+Y_{n-1}X_0$$

Estas son las fórmulas que hemos usado en la hoja de cálculo.

Puedes consultar la búsqueda de la primera solución por fracciones continuas y la recurrencia para las siguientes en las hojas de cálculo

http://www.hojamat.es/sindecimales/aritmetica/herramientas/hoja/pell.ods

http://www.hojamat.es/sindecimales/aritmetica/herramientas/hoja/pell.xls

Por ejemplo, intenta resolver esta cuestión: ¿Qué cuadrado perfecto de diez cifras, al quitarle una unidad se puede descomponer en cinco cuadrados perfectos idénticos?

Algoritmo 196

El número 98 necesita 24 iteraciones para desembocar en capicúa.

El número 110 desemboca en el 121

Es fácil razonar que la segunda cifra de un capicúa de tres cifras que sea meta ha de ser par.

En efecto, si proviene de un número de dos cifras ha de ser múltiplo de 11, y los únicos capicúas son 121, 242, 363 y 484, todos con las decenas pares.

Si proviene de un número de tres cifras 100a+10b+c, tendrá la forma 101(a+c)+20b, con a+c<=9 y b<6, con lo que sus cifras serán a+c, 2b, a+c, y por tanto la cifra de las decenas será par.

Las metas de cuatro cifras han de ser múltiplos de 11. Es evidente por ser capicúas N=1000a+100b+100b+a=1001a+110b

Suma de tres números triangulares

La solución es el número 952, que admite estas 24 descomposiciones en triangulares:

0	6	946
0	91	861
1	171	780
1	210	741
3	3	946
6	351	595
10	276	666
21	28	903
21	190	741
21	435	496
36	55	861
36	136	780
55	231	666

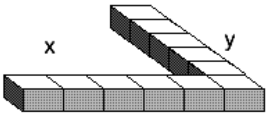
66	66	820
66	325	561
78	171	703
78	378	496
91	120	741
91	231	630
91	300	561
105	351	496
136	351	465
171	253	528
276	325	351

Factorización de Fermat

(1) La variable **y** ha de estar acotada, porque tanto **x+y** como **y-x** son menores que N (al menos uno de ellos), luego su suma cumplirá **2y<2N** y por tanto N es una cota para y.

(2) Para saber si un número es un cuadrado sin usar la raíz cuadrada basta ir sumando los números impares consecutivos. Si una de las sumas coincide con ese número, será cuadrado perfecto. Si las sumas lo sobrepasan sin igualarse a él, no lo será.

(3) Si representamos N como un gnomon cuyo lado mayor es **y** mientras el lado de su hueco es **x**, vemos que en el caso extremo en el que **y-x** valga 1, **y** ha de ser menor o igual que (N+1)/2, lo que nos da una mejor cota para **y**. Lo podemos ver en la imagen, que corresponde a N=11.



APÉNDICE

FACTORIZACIÓN DE FERMAT

Entrada: Un número entero n

Operación: Descompone n en dos factores según la descomposición factorial sugerida por Fermat

Código en Basic

```
Sub factorfermat(n)

dim y,a,m,x,j,s,b,p as long

'Se hace crecer la variable "a" hasta que sobrepase a "n"

a=1:y=1:i=1

while a<n
i=i+2:a=a+i:y=y+1’la variable  es suma de impares consecutivos y por tanto cuadrado perfecto
wend

' Aquí se resuelve el caso en el "n" sea cuadrado perfecto

if a=n then
msgbox(y)
msgbox(y)
i=i+2:a=a+i:y=y+1
end if

' Se hacen crecer "x" e "y"


x=1:j=1:b=1

while y+y<=n+1
while n+b<a
j=j+2
b=b+j
x=x+1
```

```
wend

'Se ha descubierto una coincidencia

if a=n+b then
m=x+y
p=y-x
msgbox(m)
msgbox(p)
end if

i=i+2:a=a+i:y=y+1

wend
End Sub
```

FRACCIONES CONTINUAS

Ecuación de Pell

Soluciones: 2687489281