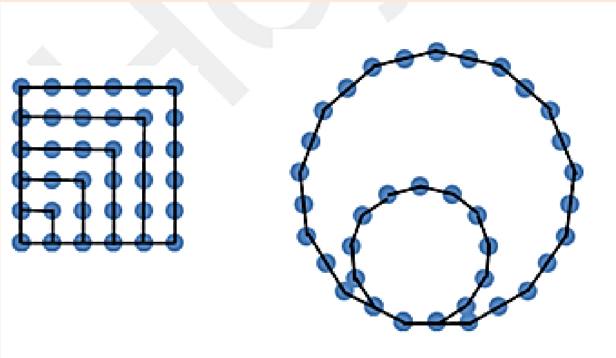
Números y hoja de cálculo XII



Curso 2019-20

Colección Hojamat.es

http://www.hojamat.es

PRESENTACIÓN

Esta publicación recoge el resumen de lo publicado en mi blog “Números y hoja de cálculo”

(http://hojaynumeros.blogspot.com/)

durante el curso 2019-20.

En esta ocasión no contiene ningún ciclo de temas, como ocurrió en otras ocasiones con los números piramidales, los cuadrados de cifras o los números AROLMAR. En realidad, los apartados que figuran en esta publicación se podían haber organizado de otra forma. El más claro es el de Números especiales, pero ha sido fruto de la casualidad y no de una planificación.

Para el curso próximo sí contaremos con ciclos de temas, con lo que este quedará como una cierta excepción.

El autor no esperaba llegar a doce resúmenes anuales, ante la posible escasez de nuevos temas, pero lo leído diariamente en Twitter ha significado una muy buena fuente de inspiración.

CONTENIDO

SIGUE EL MISMO TIPO AL DUPLICAR LAS UNIDADES

El número 144 es cuadrado, $144=12^2$, y si duplicamos su última cifra resulta otro cuadrado, pues $1444=38^2$

¿Ocurrirá esto con otros cuadrados? ¿Existirán ejemplos similares con números primos, triangulares y de otro tipo? Lo estudiamos.

Cuadrados

Los únicos cuadrados que presentan duplicadas sus dos últimas cifras son los terminados en 44 o en 00. No existirán casos con otras cifras. Lo vemos detenidamente:

Las terminaciones de los números cuadrados son 0, 1, 4, 5, 6 y 9. En los casos 1, 5, 6 y 9 es imposible la terminación en 11, 55, 66 o 99. En todos los razonamientos llamaremos **a** a la cifra de las decenas de la posible raíz cuadrada.

Un número terminado en 1 o en 9 no puede producir un cuadrado terminado en 11, pues si termina en **a1**, su cuadrado lo hará en (2a)1, y 2a no puede valer 1, y si termina en 9, el cuadrado de **a9** terminaría en (18a+8)1, y tampoco podría terminar en 11. Desechamos, pues la terminación 11.

La 55 tampoco es posible terminación de cuadrado, pues si un número termina en **a5**, su cuadrado lo hará en (10a+2)5, y el paréntesis par no puede producir un 5 en las decenas.

Para producir un 66 la raíz cuadrada ha de terminar en **a6** o en **a4**. En el primer caso el cuadrado terminaría en (12a+3)6, y el paréntesis no puede terminar en 6. En el otro caso sería (8a+1)6, que tampoco produce 66.

La 99 provendría de un número terminado en **a3** o en **a7**, y su cuadrado terminaría en (6a)9 en el primer caso y (14a+4)9 en el segundo, lo que imposibilita el 99 como terminación.

La terminación en 00 para un cuadrado provendría de una raíz cuadrada terminada en 0. Hasta aquí bien, pero para la cuestión que nos ocupa debería también ser un cuadrado, con lo que tendría un número par de ceros, y al añadirle otro cero sería un número impar, que no podría ser cuadrado.

Por tanto, la única duplicación de unidades que produce un cuadrado es la 44. Si deseamos más casos además del 144 deberemos buscar entre los cuadrados terminados en 4.

Búsqueda de cuadrados del tipo dado

Lo iniciaremos en Basic de Excel para abordar el tema y extenderlo más tarde a otros casos. Usaremos la función *sigueigual*, que iremos adaptando a lo largo del estudio. Para cuadrados puede ser esta:

Public Function sigueigual(n) as boolean
Dim a, c

c = n Mod 10 ‘Encuentra la cifra de las unidades
If c <> 4 Then sigueigual = False: Exit Function ‘Si no termina en 4, lo dejamos
a = n * 10 + c ‘Formamos la duplicación de las unidades
If escuad(n) And escuad(a) Then sigueigual = True Else sigueigual = False
‘Si el número es cuadrado antes y después de duplicar, vale
End Function

La función *escuad* puede tener este código:
Public Function escuad(n) As Boolean
'Determina si n es un cuadrado

If n < 0 Then
escuad = False
Else
If n = Int(Sqr(n)) ^ 2 Then escuad = True Else escuad = False

End If
End Function

Probamos los primeros números con esta función y solo nos resulta la solución 144 y 1444, por lo que necesitamos una herramienta más potente, como el lenguaje PARI. Usaremos solo cuadrados, con lo que la búsqueda será más rápida. El listado que usaremos es este:

for(i=1, 1000000000, n=i*i; q=n%10; if(q==4, r=q+n*10; if(issquare(r), print1(n,", ",sqrtint(n),", ",r,", ",sqrtint(r)))))

Para cada valor de la variable **i** forma su cuadrado **n**. Si termina en 4, se duplica la cifra de las unidades para formar la variable **r** y si es un cuadrado, hemos encontrado la solución. Con este código aparece la solución 144, 1444, y también dos más:

144, *12*, 1444, *38*, 432374632704, *657552*, 4323746327044, *2079362*, 899063381008862784, *948189528*, 8990633810088627844, *2998438562,…*

(Escribimos en cursiva las raíces cuadradas del término anterior)

Vemos que son escasos los cuadrados con esta propiedad. Con un poco de paciencia se podrían buscar más soluciones, pero con las tres dadas se advierte su rareza.

Estos números (los que son raíces cuadradas del segundo cuadrado, como 2998438562) pertenecen a la sucesión http://oeis.org/A239364, que son soluciones de la ecuación de Pell $x^2-10y^2=4$, que viene a exigir que al añadir un 4 a un cuadrado y^2 se convierta en otro cuadrado x^2, pero en la sucesión indicada figuran otras soluciones, que son las que no terminan en 44.

Primos

Podemos ir adaptando la función *sigueigual* según el tipo de números que estudiemos. En el caso de los primos podría ser:

```
Public Function sigueigual(n)
Dim a, c

c = n Mod 10
a = n * 10 + c
If esprimo(n) And esprimo(a) Then sigueigual = True
Else sigueigual = False
End Function
```

Resultan estos primeros ejemplos, que, como vemos, son mucho más frecuentes que los cuadrados:

19, 23, 31, 43, 59, 67, 73, 97, 103, 127, 139, 149, 151, 173, 181, 193, 199, 211, 233, 239, 241, 263, 269, 271, 277, 283, 349, 353, 367, 373, 383, 409, 421, 479, 487, 499, 509, 523, 547, 571, 601, 613, 619, 631,…

Por ejemplo, 173 es primo y 1733 también.

Con PARI basta con un código muy simple:

forprime(n=2, 2000, p=(n%10)+n*10; if(isprime(p), print(n,", ")))

Puedes experimentar con él aumentando el rango de búsqueda, que en el listado va de 2 a 2000. Observa lo útil que es la instrucción ***forprime***.

Triangulares

En el caso de los triangulares volvemos a la escasez de resultados. En la siguiente versión de *sigueigual* usamos la condición para que **n** sea triangular, y es que 8*n+1 sea cuadrado:

```
Public Function sigueigual(n)
Dim a, c

c = n Mod 10
a = n * 10 + c
If escuad(8 * n + 1) And escuad(8 * a + 1) Then sigueigual = True Else sigueigual = False
End Function
```

En una primera búsqueda obtenemos cuatro soluciones: 6, 66, 171 y 1540.

Para encontrar otros ejemplos necesitamos usar PARI:

for(i=1, 10000000000, n=i*(i+1)/2;q=n%10;r=q+n*10; if(issquare(8*r+1), print1(n,", ")))

En primer lugar construimos un triangular mediante su definición, *n=i*(i+1)/2*, y después le adosamos el último dígito y comprobamos que sigue siendo triangular mediante la prueba *issquare(8*r+1).*

De esta forma obtenemos más soluciones:

6, 66, 171, 1540, 21454525, 43809480, 1395379509846, 5671003058155, 337549427259780, 3969358565670798 6,…

No son tan escasos como los cuadrados, pero se ve que aparecerán de forma aislada.

Oblongos

Ya que hemos recorrido los tipos más estudiados, completamos con alguno más. Por ejemplo, con los oblongos.

Como estos números son dobles de un triangular, el criterio del 8*n+1 que estudiamos anteriormente se modifica en que sea cuadrada la expresión 4*n+1. Así quedaría *sigueigual*:

Public Function sigueigual(n)
Dim a, c
c = n Mod 10
a = n * 10 + c

If escuad(4 * n + 1) And escuad(4 * a + 1) Then sigueigual = True Else sigueigual = False
End Function

Las dos primeras soluciones que nos da esta función, 342 y 3080, resultan ser dobles de dos soluciones para triangulares, como son 171 y 1540.

Si ampliamos usando PARI comprobamos que estos ejemplos son también escasos:

342, 3080, 225150, 87618960, 711635652, 6404720870, 182191536189390, 675098854519560,…

Con esto ya tenemos una idea de lo que da de sí esta cuestión. Lo dejamos aquí.

MÚLTIPLOS ANAGRAMÁTICOS

El estudio que sigue se basa en un "twitt" publicado en nuestra cuenta de Twitter (@connumeros) el 11/6/19.

El número de fecha de hoy, 11619, presenta dos múltiplos sencillos anagramáticos (mismas cifras en distinto orden):

*11619*13 = 151047*

*11619*9 = 104571*

Se puede sospechar que todo número puede cumplir el tener dos múltiplos anagramáticos para factores convenientemente grandes, pero como no tenemos herramientas para este tipo de búsquedas nos limitaremos a múltiplos que usan factores no mayores que un número determinado. Después nos podemos plantear una extensión del estudio.

Para investigar esta cuestión necesitamos una función que nos indique si dos números poseen las mismas cifras y con la misma frecuencia pero en distinto orden. Contamos con dos versiones de esta función. La primera, de hace años, se basa en la conversión de cada número en cadena de texto, para después investigar si cada carácter de uno se encuentra en el otro. Las frecuencias las tiene en cuenta borrando cada carácter encontrado en ambas cadenas. La puedes consultar en una entrada de 2012 de nuestro blog:

http://hojaynumeros.blogspot.com/2012/04/proposito-de-ormiston.html

Aquí usaremos otra versión, que no necesita funciones de texto, pero sí varias memorias que almacenen las cifras de cada uno de los números.

Public Function cifras_identicas(m, n) As Boolean
Dim i, h, s
Dim ci As Boolean
Dim ca(10), cb(10)

```
For i = 1 To 10: ca(i) = 0: cb(i) = 0: Next i ‘Prepara memorias para recibir frecuencias de cifras

h = m ‘Extracción de cifras del primer número
While h > 0
i = Int(h / 10)
s = h - i * 10
h = i
ca(s + 1) = ca(s + 1) + 1 ‘Almacenamiento de las frecuencia de una cifra
Wend

h = n ‘El mismo proceso para el otro número
While h > 0
i = Int(h / 10)
s = h - i * 10
h = i
cb(s + 1) = cb(s + 1) + 1
Wend

ci = True
For i = 1 To 10
If ca(i) <> cb(i) Then ci = False ‘Si una de las frecuencias no coincide, las cifras no son idénticas
Next i
cifras_identicas = ci
End Function
```

Con esta función podemos encontrar dos múltiplos de un número que presenten las mismas cifras.

En PARI es mucho más simple, pues permite la ordenación de los dígitos de un número en forma de vector:

anagram(a,b)=vecsort(digits(a))==vecsort(digits(b))

Listado de soluciones

Podemos investigar si un número dado posee dos múltiplos anagramáticos al menos. Tal como explicamos más arriba, solo buscaremos múltiplos hasta un factor dado, para no alargar ni bloquear el proceso. Hemos creado la función *multianagram*, que busca los múltiplos anagramáticos hasta una cota k.

Como lo que nos interesa es saber si existe solución o no, está construida de forma que detiene el proceso cuando encuentra la primera solución. Sus parámetros son **n**, el número dado, y **k**, la cota para los factores que formarán los múltiplos pedidos.

Public Function multianagram(n, k) As String
Dim s$
Dim i, j, a, b, l

s$ = "" ‘Si no hay solución, devuelve un string vacío.
i = 2
While i <= k And s$ = "" ‘Recorre los factores hasta una cota ***k***

j = 1
While j < i And s$ = "" ’Busca el segundo múltiplo
If i <> j Then
a = n * i: b = n * j
If cifras_identicas(a, b) Then s$ = s$ + Str$(i) + Str$(j) + Str$(a) + Str$(b)
‘Si las cifras son idénticas, devuelve los dos factores y los múltiplos resultantes
End If
j = j + 1
Wend
i = i + 1
Wend
multianagram = s$
End Function

Hemos preparado varias búsquedas con la cota 1000 para los factores que producirán los múltiplos anagramáticos. Parece que todos los primeros números presentan la propiedad.

Por ejemplo, entre 30 y 45, todos poseen un par al menos de múltiplos anagramáticos. En el listado figura cada número, después el par de factores necesarios y al final los dos múltiplos pedidos, que presentan las mismas cifras:

30	7 4 210 120
31	42 33 1302 1023
32	57 39 1824 1248
33	7 4 231 132
34	48 39 1632 1326
35	55 37 1925 1295
36	5 3 180 108
37	13 4 481 148
38	37 28 1406 1064
39	31 28 1209 1092
40	21 12 840 480
41	61 25 2501 1025
42	22 7 924 294
43	41 32 1763 1376
44	38 29 1672 1276
45	7 3 315 135

Modificando ligeramente la función *multianagram* podemos tener una idea del máximo factor necesario para lograr el par de múltiplos anagramáticos. Para los número del 1 al 100, el máximo es 137, necesario para el número 95 y sus múltiplos 95*137=13015 y 95*119=11305.

Hasta el 500 el número que necesita un factor mayor es el 425, que forma los múltiplos 12750 y 70125 con un factor máximo de 165. De este orden de magnitud suelen ser los factores. Por eso, para cota 1000 todos los primeros números poseen múltiplos anagramáticos.

Con PARI podemos investigar qué factor presenta cada número de los dados (que por ahora son todos). Con este listado se logra:

```
anagram(a,b)=vecsort(digits(a))==vecsort(digits(b))
```

```
for(n=200,500,k=2;e=0;while(k<=200&&e==0,h=1;while(h<=k-1&&e==0,a=n*k;b=n*h;e=anagram(a,b);if(e<>0,print(n,", ",k));h+=1);k+=1))
```

En primer lugar definimos anagram(a,b), que devuelve 1 si los números son anagramáticos y 0 si no lo son. Después construimos los múltiplos posibles e investigamos si existen anagramáticos y cuál es el factor máximo. En el ejemplo buscamos entre 2 y 200 con un factor máximo de 200 (Observa ***k<=200***) Después se pueden cambiar los parámetros. Estos serían los resultados desde 420 a 430:

420, 22
421, 65
422, 74
423, 4
424, 84
425, 165
426, 58
427, 45
428, 147
429, 47

430, 41

Comprobamos que 425 es el que necesita un factor mayor (dentro de la cota 200)

Problema contrario

Podemos plantear el problema contrario, y es la búsqueda de números que no posean múltiplos anagramáticos para una cota razonable, como puede ser 1000. Si encontramos alguno, subimos la cota.

Podemos usar nuestro buscador en Excel o el código anterior de PARI ligeramente modificado.

```
anagram(a,b)=vecsort(digits(a))==vecsort(digits(b))

for(n=2,5000,k=2;e=0;while(k<=1000&&e==0,h=1;while(h<=k-1&&e==0,a=n*k;b=n*h;e=anagram(a,b);h+=1);k+=1);if(e==0,print(n)))
```

En este caso solo imprimimos resultados si después de recorrer los múltiplos no se encuentra ningún par anagramático. Usamos cota 1000, a ver qué ocurre. Hasta 5000 no hemos encontrado ninguno. Vamos aumentando el rango de búsqueda, aunque se va lentificando el proceso. Hasta una cota de 100000, con un factor máximo de 1000, no ha aparecido ningún caso. Es arriesgado conjeturar nada, porque si aumenta el factor, también lo hace el número de cifras del múltiplo, lo que dificulta la coincidencia. Lo dejamos aquí.

NÚMEROS DE SMITH DE SEGUNDO ORDEN

Este concepto de “número de Smith de segundo orden” no parece que se use mucho, porque no figura como tal en las páginas tipo Wikipedia o Mathworld. Hemos tomado esta notación de la página

http://oeis.org/A174460

En realidad, en esa sucesión, como veremos más adelante, no figuran todos los números que consideraremos aquí.

Los números que sí son populares y figuran en muchas publicaciones son los números de Smith de primer orden (http://oeis.org/A006753). Son aquellos en los que la suma de las cifras del número coincide con la suma del mismo cálculo en sus factores primos. Albert Wilansky los nombró números de Smith por su cuñado Harold Smith que tenía el número de teléfono 4937775 y presentaba esta propiedad, es decir

4937775 = 3.5.5.65837

y

4 + 9 + 3 + 7 + 7 + 7 + 5 = 3 + 5 + 5 + (6 + 5 + 8 + 3 + 7) = 42

Puedes encontrar mucho material sobre ellos en la página http://www.shyamsundergupta.com/smith.htm
Los números que proponemos aquí poseen la misma propiedad pero con cuadrados. Así, 822=2*3*137, y la suma de los cuadrados de las cifras de 822 coincide con la suma de las de sus factores:

8^2+2^2+2^2 = 72 y 2^2+3^2+1^2+3^2+7^2=72.

Puedes encontrar muchos de ellos en la sucesión http://oeis.org/A174460, aunque en ella se exige la no coincidencia de dígitos, que no consideraremos aquí.

Búsqueda de números de Smith de segundo orden

Podemos usar la factorización de un número y nuestra función *sumacifras(n;2).* Hemos creado la función *sum2factores*, que encuentra los factores primos de un número con multiplicidad y va tomando nota de la suma de sus cifras al cuadrado:

Public Function sum2factores(n)
Dim f, a, e

a = n ‘Copia el valor de n

f = 2 ‘Recogerá los factores primos

e = 0 ‘Recogerá la suma de las cifras al cuadrado

While f <= a

While a / f = a \ f ‘Ve si es divisor

a = a / f: e = e + sumacifras(f, 2)’ Si lo es, suma los cuadrados de las cifras

Wend

If f = 2 Then f = 3 Else f = f + 2 ‘Siguiente factor, que será primo por la división a=a/f

Wend

sum2factores = e ‘Suma total

End Function

Con esta función basta buscar los números en los que coinciden las sumas de los cuadrados de las cifras, tanto las propias como las de los factores. Sólo buscaremos entre los números compuestos, como era de esperar.

Los primeros resultados son

56	61	[2,3][7,1]
58	89	[2,1][29,1]
810	65	[2,1][3,4][5,1]
822	72	[2,1][3,1][137,1]
1075	75	[5,2][43,1]
1111	4	[11,1][101,1]
1255	55	[5,1][251,1]
1519	108	[7,2][31,1]
1752	79	[2,3][3,1][73,1]
2145	46	[3,1][5,1][11,1][13,1]
2227	61	[17,1][131,1]
2260	44	[2,2][5,1][113,1]
2483	93	[13,1][191,1]
2618	105	[2,1][7,1][11,1][17,1]
2620	44	[2,2][5,1][131,1]
3078	122	[2,1][3,4][19,1]
3576	119	[2,3][3,1][149,1]
3653	79	[13,1][281,1]
3962	130	[2,1][7,1][283,1]
4336	70	[2,4][271,1]
4823	93	[7,1][13,1][53,1]
4974	162	[2,1][3,1][829,1]

En la tabla figuran, en la segunda columna, las sumas de cuadrados de cifras que son coincidentes. Por ejemplo, en 4823 figura la suma 93 y los factores 7, 13, 53 (los segundos números son exponentes), y, efectivamente:

4^2+8^2+2^2+3^2=93 y 7^2+1^2+3^2+5^2+3^2=93

En forma de listado:

56, 58, 810, 822, 1075, 1111, 1255, 1519, 1752, 2145, 2227, 2260, 2483, 2618, 2620, 3078, 3576, 3653, 3962, 4336, 4823, 4974, 5216, 5242, 5386, 5636, 5719, 5762, 5935, 5998, 6220, 6424, 6622, 6845, 7015, 7251, 7339, 7705, 7756, 8460, 9254, 9303, 9355,…

Están repartidos en dos sucesiones,

http://oeis.org/A174460
A174460 Smith numbers of order 2. 7 56, 58, 810, 822, 1075, 1519, 1752, 2145, 2227, 2260, 2483, 2618, 2620, 3078, 3576, 3653, 3962, 4336, 4823, 4974, 5216, 5242, 5386, 5636, 5719, 5762, 5935, 5998, 6220, 6424, 6622, 6845, 7015, 7251, 7339, 7705, 7756, 8460, 9254, 9303, 9355, 10481, 10626, 10659

http://oeis.org/A176670
1111, 1255, 12955, 17482, 25105, 28174, 51295, 81229, 91365, 100255, 101299, 105295, 107329, 110191, 110317, 117067, 124483, 127417, 129595, 132565, 137281, 145273, 146137, 149782, 163797, 171735, 174082, 174298, 174793

Nosotros unificamos las dos, pues nos parece más sencillo y directo.
Podemos acudir también al lenguaje PARI:

sum2cifras(n)=norml2(digits(n))
sum2factor(n)=local(f, s=0); f=factor(n); for(i=1, matsize(f)[1], s+=sum2cifras(f[i, 1])*f[i,2]); s

for(i=1,10000,if(sum2cifras(i)==sum2factor(i)&&isprime(i)==0,print1(i,", ")))

Es una traducción de nuestro procedimiento. Con este código se repiten los resultados:

```
56, 58, 810, 822, 1075, 1111, 1255, 1519, 1752, 2145, 2227, 2260, 2483, 2618, 26
20, 3078, 3576, 3653, 3962, 4336, 4823, 4974, 5216, 5242, 5386, 5636, 5719, 5762
, 5935, 5998, 6220, 6424, 6622, 6845, 7015, 7251, 7339, 7705, 7756, 8460, 9254,
9303, 9355,
```

Casos particulares

Dentro de este listado podemos investigar algunos tipos particulares de números, que coinciden con los que solemos considerar en estos documentos.

Semiprimos

Estos son los primeros números de Smith de segundo orden que son semiprimos. Para encontrarlos basta añadir la condición de que posean solo dos factores primos.

58	89	[2,1][29,1]
1111	4	[11,1][101,1]
1255	55	[5,1][251,1]
2227	61	[17,1][131,1]
2483	93	[13,1][191,1]
3653	79	[13,1][281,1]
5242	49	[2,1][2621,1]
5386	134	[2,1][2693,1]
5935	140	[5,1][1187,1]
5998	251	[2,1][2999,1]
7251	79	[3,1][2417,1]
7339	148	[41,1][179,1]
9355	140	[5,1][1871,1]

En Excel hemos añadido la condición *essemiprimo* y en PARI con bigomega(n)==2:

for(i=1,100000,if(sum2cifras(i)==sum2factor(i)&&isprime(i)==0&&bigomega(i)==2,print1(i,", ")))

58, 1111, 1255, 2227, 2483, 3653, 5242, 5386, 5935, 5998, 7251, 7339, 9355, 10481, 12381, 12813, 12955, 14359, 16586, 16658, 16757, 16941, 17349, 17482, 18215, 18273, 18566, 19271, 19641, 21815, 23867, …

Cuadrados

Los números de Smith de segundo orden que son cuadrados son mucho menos abundantes. Estos son los primeros:

209764	186	[2,2][229,2]
476100	102	[2,2][3,2][5,2][23,2]
729316	180	[2,2][7,2][61,2]
908209	230	[953,2]
1081600	102	[2,8][5,2][13,2]
6230016	86	[2,12][3,2][13,2]
9909904	340	[2,4][787,2]
11404129	120	[11,2][307,2]
11799225	246	[3,2][5,2][229,2]
12068676	226	[2,2][3,4][193,2]
13242321	48	[3,2][1213,2]
15007876	224	[2,2][13,2][149,2]
23765625	188	[3,2][5,6][13,2]
29170801	200	[11,2][491,2]
31933801	174	[5651,2]
39601849	288	[7,2][29,2][31,2]
40627876	254	[2,2][3187,2]
41563809	232	[3,2][7,2][307,2]
42902500	130	[2,2][5,4][131,2]
49449024	230	[2,6][3,2][293,2]
54434884	226	[2,2][7,2][17,2][31,2]

Por ejemplo, 476100 es el cuadrado de 690, y se cumple:

4^2+7^2+6^2+1^2+0^2+0^2=102
690=2*3*5*23
2^2+3^2+5^2+2^2+3^2+2^2+3^2+5^2+2^2+3^2=102

Tienen la ventaja algorítmica de que en los bucles FOR-NEXT podemos usar ***i*i*** en lugar de ***i***, con lo que los cálculos son mucho más rápidos.

En PARI

```
for(i=1,30000,n=i*i;a=norml2(digits(n));
s=0;f=factor(n); for(k=1, matsize(f)[1],
s+=sum2cifras(f[k, 1])*f[k,2]);if(a==s,print1(n,", ")))
```

209764, 476100, 729316, 908209, 1081600, 6230016, 9909904, 11404129, 11799225, 12068676, 13242321, 15007876, 23765625, 29170801, 31933801, 39601849, 40627876, 41563809, 42902500, 49449024, 54434884, 61121124, 78216336, 78801129, 103795344, 110838784, 116380944, 128595600, 132917841, 134026929, 137569441, 151363809, 158206084, 161976529, 163175076,

Triangulares

Para encontrar los triangulares, basta organizar un bucle con i*(i+1)/2. Resultan los siguientes:

3	9	[3,1]
2145	46	[3,1][5,1][11,1][13,1]
163306	91	[2,1][11,1][13,1][571,1]
191271	137	[3,1][103,1][619,1]
248160	121	[2,5][3,1][5,1][11,1][47,1]
561270	115	[2,1][3,1][5,1][53,1][353,1]
591328	184	[2,5][17,1][1087,1]
860016	137	[2,4][3,1][19,1][23,1][41,1]
1017451	93	[23,1][31,1][1427,1]
1296855	236	[3,2][5,1][7,1][23,1][179,1]
1355481	141	[3,3][61,1][823,1]
2372931	157	[3,2][11,2][2179,1]
2713285	156	[5,1][17,1][137,1][233,1]
3386503	152	[19,1][137,1][1301,1]
3517878	261	[2,1][3,1][7,1][13,1][17,1][379,1]
3755170	158	[2,1][5,1][137,1][2741,1]
3899028	303	[2,2][3,1][7,2][19,1][349,1]
3932610	140	[2,1][3,1][5,1][11,1][17,1][701,1]
4074085	170	[5,1][571,1][1427,1]
4247155	136	[5,1][11,1][31,1][47,1][53,1]
4683330	143	[2,1][3,2][5,1][17,1][3061,1]

Con PARI:

```
sum2cifras(n)=norml2(digits(n))
sum2factor(n)=local(f, s=0); f=factor(n); for(i=1, matsize(f)[1], s+=sum2cifras(f[i, 1])*f[i,2]); s
for(i=1,10000,n=i*(i+1)/2;if(sum2cifras(n)==sum2factor(n),print1(n,", ")))
```

3, 2145, 163306, 191271, 248160, 561270, 591328, 860016, 1017451, 1296855, 1355481, 2372931, 2713285, 3386503, 3517878, 3755170, 3899028, 3932610, 4074085, 4247155, 4683330, 4717056, 4750903, 5076891, 5546115, 5680135, 6917340, 6928503, 7172578, 7571886, 8547045, 8650720,…

Cubos

Por último, como ejemplo de búsquedas que se pueden emprender, los primeros cubos que son números de Smith de segundo orden. Dejamos como ejercicio su obtención.

1601613	84	[3,6][13,3]	
76225024	138	[2,9][53,3]	
1302170688	228	[2,6][3,3][7,3][13,3]	
1556862679	357	[19,3][61,3]	
1745337664	246	[2,6][7,3][43,3]	

Con PARI:

1601613, 76225024, 1302170688, 1556862679, 1745337664, 4427101288, 14331199589, 36198994112, 47124116928, 48149090072, 56006043976, 57289251375, 64481201000, 66037242088, 72772859375, 96955616584, 149467669443, 197849544283, 262266899201, 262389836808, 386464682304, 484661551375, 574270627328, 761048497000, 795052191744, 841499753121,…

Aquí dejamos los casos. Puedes seguir investigando en esta dirección.

RELACIONES ENTRE POTENCIAS

Surgen nuevas coincidencias de sumas de cuadrados de cifras si comparamos cualquier número con sus potencias o bien entre ellas. Es inevitable que este estudio conste de un conjunto de casos, ya que no hay una teoría general que lo respalde y hay que acudir a búsquedas un poco al azar.

Cifras de un número y su cuadrado

Normalmente, salvo las potencias de 10 y sus múltiplos más pequeños, no se perciben relaciones entre las cifras de un número y las de su cuadrado. Por ello, debemos acudir a las búsquedas. En este tema usaremos la función *sumacifras* con parámetro 2. Esta función ya la conoces si has leído los temas anteriores de esta serie.

Comenzaremos buscando números cuya suma de cuadrados de cifras coincida con la de su cuadrado.

Cuadrado con la misma suma

Si exigimos en Excel

s*umacifras(N;2)=sumacifras(N^2;2)*

o en PARI

norml2(digits(n))==norml2(digits(n^2))

obtendremos los números buscados. Son estos:

1	1	1
100	1	10
1225	34	35
10000	1	100
23104	30	152
122500	34	350
142129	107	377
204304	45	452
290521	115	539
502681	130	709
1000000	1	1000
1687401	167	1299
1954404	155	1398
2070721	107	1439
2307361	108	1519
2310400	30	1520
2461761	143	1569
2531281	108	1591
2819041	167	1679
3861225	143	1965
6754801	191	2599
8054244	141	2838

Figuran en la primera columna los cuadrados, en la segunda la suma común, y en la tercera el número buscado.

Con PARI usamos el código

for(i = 1, 30000, if(norml2(digits(i^2)) == norml2(digits(i)), print1(i, ", ")))

Resulta el mismo listado que con Excel.

1, 10, 35, 100, 152, 350, 377, 452, 539, 709, 1000, 1299, 1398, 1439, 1519, 1520, 1569, 1591, 1679, 1965, 2599, 2838, 3332, 3500, 3598, 3770, 4520, 4586, 4754, 4854, 5390, 5501, 5835, 5857, 6388, 6595, 6735, 6861, 6951, 7090, 7349, 7887, 8395, 9795, 10000, 10056,

Esta sucesión estaba inédita y la hemos publicado en https://oeis.org/A309883.

Puedes comprobar que si K pertenece a la sucesión, K*10^p, con p entero positivo, también pertenece a la misma.

Como curiosidad, entre ellos hay capicúas, como 7887, 67476 y 93939.

También se encuentran números primos; 709, 1439, 5501, 5857, 7349, 10159, 14897, 17449, 25999,…

Los únicos términos menores que 100000 de la sucesión de Fibonacci son 1 y 377.

Así podríamos seguir.

Suma del cuadrado es el doble

Ahora veremos los números que son cuadrados y la suma de los cuadrados de sus cifras es el doble que la correspondiente a su raíz cuadrada. Por ejemplo:
504^2 = 254016.

Los cuadrados de las cifras de 504 suman 5^2 + 0^2 + 4^2 = 41.

La suma en su cuadrado es 2^2 + 5^2 +4^2 + 0^2 + 1^2 +6^2 = 82, que es el doble de 41.

Los primeros números que cumplen esto son:

72, 405, 504, 720, 722, 953, 1964, 2092, 2376, 2555, 2577, 2619, 4005, 4050, 4284, 4449, 4571, 5004, 5040, 5552, 5651, 5805, 6326, 6615, 7200, 7218, 7220, 7676, 8355,…

También aquí si K pertenece a la sucesión, K*10^p, con p entero positivo, también pertenece a la misma.

Puedes plantearte como ejercicio un código PARI para este caso.

64146 es el primer capicúa que cumple la condición:
64146^2=4114709316

6^2+4^2+1^2+4^2+6^2=105
4^2+1^2+1^2+4^2+7^2+0^2+9^2+3^2+1^2+6^2=210, que es el doble de 105.

Como en el caso anterior, aparecen primos:
953, 5651, 18251, 19913, 20129, 22691, 33587, 34487,…

Cubos con igual suma

De forma similar, podemos buscar aquellos números que coinciden con su cubo en la suma de los cuadrados de las cifras.

Adaptando las técnicas que usamos para los cuadrados, es fácil ver que los primeros con ese tipo de coincidencia son los de la tabla siguiente:

1	1	1
1000	1	10
405224	65	74
1000000	1	100
405224000	65	740
1000000000	1	1000

La primera columna está formada por los cubos, la tercera por las bases y la central por la suma común.

El listado de los primeros es el siguiente:

1, 10, 74, 100, 740, 1000, 3488, 7400, 10000, 23658, 30868, 34880, 47508, 48517, 52187, 58947, 59468, 67685, 68058, 74000, 76814, 78368, 78845, 84878, 100000

También esta sucesión estaba inédita, y la hemos publicado en https://oeis.org/A309884.

Se puede engendrar en PARI con:

for(i = 1, 10^6, if(norml2(digits(i^3)) == norml2(digits(i)), print1(i, ", ")))

Cubos con suma triple

Adjuntamos los resultados sin explicación:

438976	255	76
43243551	105	351
438976000	255	760
1589324463	261	1167

76, 351, 760, 1167, 1338, 1717, 2494, 2822, 3184, 3510, 5056, 5827, 6092, 6166, 6183, 7046, 7175, 7261, 7600, 8162, 8306, 8347, 8842, 8853, 9862, 10064, 10067, 10447, 11048, 11285, 11559, 11670, Por ejemplo, 76^3=438976, y se cumple:

7^2+6^2=85
4^2+3^2+8^2+9^2+7^2+6^2=255=85*3

En PARI:

for(i = 1, 2*10^4, if(norml2(digits(i^3)) == 3*norml2(digits(i)), print1(i, ", ")))

Cuadrados con cubos

Una vez abierto el camino, no es interesante recorrer muchos casos, por lo que paramos aquí. Estos son los primeros números en los que coinciden las sumas de cuadrados de las cifras en el cuadrado y el cubo:

1, 10, 100, 1000, 4313, 4847, 5407, 5911, 6856, 7414, 8366, 9317, 10000, 10949, 12234, 13514, 13579, 13943, 17294, 21648, 21969, 22156,…

Por ejemplo, el 4847:

4847^2=23493409 y

2^2+3^2+4^2+9^2+3^2+4^2+0^2+9^2=216

4847^3=113872553423 y

1^2+1^2+3^2+8^2+7^2+2^2+5^2+5^2+3^2+4^2+2^2+3^2=216

Con PARI

for(i = 1, 10^5, if(norml2(digits(i^2)) == norml2(digits(i^3)), print1(i, ", ")))

ALCANCES ENTRE NÚMEROS

Una cuestión amena es la de averiguar si un número de cierto tipo (primo, cuadrado, triangular,…) es alcanzable desde otro del mismo tipo mediante una recurrencia consistente en ir sumando a cada término el cuadrado de las cifras.

Por ejemplo, 23 es alcanzable desde 11, ambos primos, porque 13=11+1^2+1^2, 23=13+1^2+3^2

Todos los números sin ningún tipo especial son alcanzables, salvo los colombianos cuadráticos, ya estudiados aquí: 1, 3, 4, 5, 7, 8, 9, 10, 14, 15, 16, 18, 19, 21, 22, 25, 27, 28,…

El resto, por ejemplo el 6 y el 12 o el 13, son alcanzables desde otro número:

2+2^2=6, 3+3^2=12, 11+1^2+1^2=13.

Además, todos los son con un solo paso, pues basta elegir el último término de cualquier recurrencia.

Pasamos entonces a estudiar diversos tipos:

Números primos

Hemos preparado una función en Excel que nos puede ayudar a decidir si un número es alcanzable en el sentido que le hemos dado a la palabra. El listado que sigue es la versión entre primos, pero se cambia fácilmente por otro tipo:

Public Function alcance$(n)
'desde el final del alcance hacia atrás para ver si alguno lo alcanza
Dim i, j
Dim s$
If Not esprimo(n) Then alcance = "NO": Exit Function 'Si no es primo, renunciamos.
s$ = "" ‘Contendrá las soluciones
i = 1

```
While i <= n And s = ""
If esprimo(i) Then 'Puede ser otra condición
j = i
While j < n
j = j + sumacifras(j, 2)
If j = n Then s = s + Str$(i)
Wend
End If
i = i + 1
Wend
If s$ = "" Then s$ = "NO"
alcance = s
End Function
```

Si no existen soluciones, devuelve un "NO" y si las hay, el conjunto de ellas. Aquí tienes la función aplicada a los números primos entre 20 y 80:

23	11
29	NO
31	NO
37	NO
41	31
43	NO
47	NO
53	NO
59	NO
61	NO
67	3
71	NO
73	NO
79	NO

Solo son alcanzables 23, 41 y 67. Ampliamos el rango de búsqueda para confeccionar una tabla. Aquí tienes los primeros primos alcanzables:

Alcanzable	Base
13	11
17	3
23	11
41	31
67	3
101	19
103	19
107	2
113	19
127	71
131	73
157	2
163	73
181	71
191	109
199	11
227	31
251	3
263	89
269	7
271	211

Los puedes obtener en PARI con un algoritmo similar al de Excel:

```
forprime(n=1,1000,s=0;i=1;while(i<=n&&s==0,if(isprime(i),j=i;s=0;while(j<n&&s==0,j=j+norml2(digits(j));if(j==n,s=1;write1("final.txt",n,", "))));i+=1))
```

13, 17, 23, 41, 67, 101, 103, 107, 113, 127, 131, 157, 163, 181, 191, 199, 227, 251, 263, 269, 271, 281, 311, 379, 421, 457, 461, 467, 499, 509, 521, 541, 547, 563, …

Final de la recurrencia

Podemos plantearnos la cuestión opuesta, la de si dado un número primo, llega a alcanzar a otro primo en la recurrencia. En este tipo de cuestiones el problema que solemos tener es el de fijar una cota de búsqueda. En la sucesión A094830 se nos da una pista, pues contiene los pasos que se han de dar para cada primo.

A094830 *Start with x = n, repeatedly replace x with x + sum of squares of digits of x until you reach a prime; sequence gives number of steps.*

1, 0, 0, 6, 0, 4, 0, 11, 5, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 5, 13, 9, 0, 4, 11, 12, 17, 3, 0, 8, 0, 8, 7, 1, 7, 3, 0, 5, 7, 4, 0, 3, 0, 3, 7, 8, 0, 2, 2, 2, 6, 3, 0, 10, 2, 3, 1, 3, 0, 3, 0, 2, 2, 18, 2, 11, 0, 2, 6, 9, 0, 10, 0, 1, 1, 2, 5, 1, 0, 16, 2, 8, 0, 3, 11, 6, 10, 2, 0, 4, 1, 15, 2, 1, 9, 2, 0, 7, 7, 1

167	2633
191	2633
1709	4337
1783	4337
1831	4337
4229	10391
4421	10391
4621	10391
4787	10391
4793	10391
4817	10391
4969	10391
4993	10391
5023	10391
5231	10391
5297	10391
5623	10391
5783	10391
5849	10391
5881	10391
6073	10391
6101	10391

Según esta tabla, el máximo número de pasos de recurrencia para números pequeños es de 18, contando primos y compuestos. Podemos buscar, por ejemplo, hasta 25 iteraciones y marcar aquellos primos en los que ese número sea insuficiente.

Entre los primos menores que 5000 aparecen ya ejemplos en los que no basta con 25 iteraciones:

167, 191, 1709, 1783, 1831, 4229, 4421, 4621, 4787, 4793, 4817, 4969, 4993

Si aumento a 50, prácticamente todos los primos de ese rango alcanzan a otro primo:

Podemos conjeturar que siempre se alcanza un primo, pero actualmente no se sabe si es cierto o no.

Son muchos los que se repiten en la segunda columna. Llama la atención la frecuencia con la que aparecen. Con una nueva función que no vamos a insertar aquí, descubrimos que esos números provienen de muchos inicios primos en la recurrencia. Por ejemplo, 10391 proviene de 243 recurrencias distintas.

Números cuadrados

Los primeros cuadrados alcanzables por otro cuadrado son estos:

81	6	36
196	8	64
1024	23	529
1156	19	361
1225	4	16
1600	10	100
3600	47	2209
4489	39	1521
4624	63	3969
5625	52	2704
7056	71	5041
7225	58	3364
10201	71	5041
12100	1	1
13225	111	12321
13456	71	5041
15625	39	1521
28900	158	24964

En la primera columna el alcanzable y en la tercera el origen de la recurrencia.

Hemos adaptado la función de alcance a cuadrados

Public Function alcance$(n)
Dim i, j, m
Dim s$

If Not escuad(n) Then alcance = "NO": Exit Function
'U otra condición
s$ = ""
i = 1

```
m = 0
While i * i <= n And s = ""
'If escuad(i) Then 'puede ser otra condición, e incluso i=i
j = i * i
While j < n
j = j + sumacifras(j, 2)
If j = n Then s = s + Str$(i): m = m + 1
Wend
'End If
i = i + 1
Wend
If s$ = "" Then s$ = "NO"
alcance = s$
End Function
```

Con PARI

```
for(n=1,500,m=n*n;s=0;i=1;while(i<=n&&s==0,j=i*i;while(j<m&&s==0,j=j+norml2(digits(j));if(j==m,s=1;print1(m,", ")));i+=1))
```

Obtenemos el listado:

81, 196, 1024, 1156, 1225, 1600, 3600, 4489, 4624, 5625, 7056, 7225, 10201, 12100, 13225, 13456, 15625, 28900, 37636, 40401, 43681, 46225, 50625, 55696, …

Problema inverso

Podemos estudiar hasta donde llega cada origen. Al igual que con números primos, conjeturaremos que todos los cuadrados finalizan en otro cuadrado con un número de pasos adecuado en la recurrencia.

Hemos ido aumentando el máximo de pasos en la misma y, si llegamos a 2000 iteraciones, aún quedan estos tres al menos:

197136, 200704, 201601

Con 5000 iteraciones, los que quedan fuera se acercan a 300000:

2961841, 2965284, 2968729,…

Por último, con 10000 iteraciones, todos los cuadrados inferiores a 9000000 alcanzan a otro cuadrado. Podemos conjeturar que los cuadrados terminan alcanzando a otro cuadrado.

Triangulares

Sólo insertaremos la tabla de los primeros triangulares alcanzables por otro triangular:

Como en otros temas parecidos, una vez que hemos construido funciones para las búsquedas importantes, detenemos aquí los posibles otros casos (cubos, oblongos,…) y los dejamos como ejercicio.

NÚMEROS DE POLIGNAC

Estos números se definen a partir de la conjetura de Polignac, que pronto se descubrió que era falsa. Afirma que todo número impar es suma de un primo y de una potencia de 2. Números tan pequeños como 127 no la cumplen, por lo que duró poco como conjetura.

Llamaremos número de Polignac a aquel número impar que no cumpla la conjetura explicada, que no pueda expresarse como $p+2^x$. Se supone implícitamente que x puede valer 0, porque en ningún listado se toma el 3 como número de Polignac, ya que 3=2+2º

Son números de Polignac el 1 y el ya citado 127.

Insertaremos cuanto antes elementos de búsqueda, por lo que procede ahora el diseñar una función que nos indique si un número es de Polignac o no. No resulta difícil, porque las potencias de 2 crecen con rapidez y su cota es la llamada *valuación* del número N respecto a 2, que es el máximo exponente de una potencia de 2 que sea igual o menor que el número. Es fácil ver que se obtiene como INT(log(N)/LOG(2)). Con esa cota, vamos construyendo potencias de 2, y si al restar del número N resulta un número primo, será señal de que no es un número de Polignac.

El listado de la función puede ser el siguiente:

Function espolignac(n) as boolean
dim x
dim vale as boolean

vale=false:x=1 ‘El valor 1 es el inicio de las potencias de 2
if n/2<>n\2 then ‘Examina si el número es impar
while x<=n and not vale ‘Se recorren las potencias de 2
if esprimo(n-x) then vale=true ‘Criterio de Polignac
x=x*2 ‘Siguiente potencia de 2
wend
espolignac=not vale
else
espolignac=false ‘Si es par, no es de Polignac
end if
end function

Puedes conseguir nuestra función “*esprimo*” si buscas en Google “*función esprimo hoja*”

Con esta función es fácil encontrar números de Polignac. En la imagen tienes los primeros, obtenidos con hoja de cálculo:

1
127
149
251
331
337
373
509
599
701
757
809
877
905
907
959
977

Están publicados en http://oeis.org/A006285

A006285 Odd numbers not of form p + 2^x (de Polignac numbers).

(Formerly M5390)

1, 127, 149, 251, 331, 337, 373, 509, 599, 701, 757, 809, 877, 905, 907, 959, 977, 997, 1019, 1087, 1199, 1207, 1211, 1243, 1259, 1271, 1477, 1529, 1541, 1549, 1589, 1597, 1619, 1649, 1657, 1719, 1759, 1777, 1783, *1807, 1829, 1859, 1867, 1927, 1969, 1973 (list; graph; refs; listen; history; text; internal format)*

Esta lista se puede reproducir con el lenguaje PARI. El código propuesto en la página citada es algo difícil de entender, por lo que se puede acudir a este otro:

```
espolignac(n)={x=1;if(n/2 <> n\2,v=0;while(x<=n&&v==0, r=n-x; if(isprime(r), v=1); x=2*x);e=1-v,e=0);e}

for(n=1,1000,if(espolignac(n),print(n)))
```

Entre ellos existen primos y compuestos. También figuran en el listado algunos números impares consecutivos, como 905 y 907.

Erdös probó que existen infinitos números de este tipo, como los que tienen la forma

1260327937 + 2863311360k.

Puedes leer su fórmula en

http://www.bitman.name/math/article/388

Disponiendo de la función *espolignac* no es difícil encontrar los pares de números de Polignac consecutivos en este listado. Los primeros, menores de 10000, son estos:

905	907
3341	3343
3431	3433
4151	4153
4811	4813
4841	4843
5729	5731
7387	7389
7811	7813
8921	8923

Entre los números de Polignac, como ya se ha indicado, existen muchos primos. Los primeros son los siguientes:

127, 149, 251, 331, 337, 373, 509, 599, 701, 757, 809, 877, 907, 977, 997, 1019, 1087, 1259, 1549, 1597, 1619, 1657, 1759, 1777, 1783, 1867, 1973, 2203, 2213, 2293, 2377, 2503,...

Están publicados en http://oeis.org/A065381

Aportación nuestra

Semiprimos

También hay semiprimos entre los números de Polignac. Los primeros son:

905, 959, 1199, 1207, 1211, 1243, 1271, 1477, 1529, 1541, 1589, 1649, 1807, 1829, 1927, 1969, 1985…

Basta añadir a la condición *espolignac* la de ser semiprimo.

Con PARI podemos ampliar la lista, ya que los semiprimos se identifican porque su función bigomega es igual a 2.

```
espolignac(n)={x=1;if(n/2 <> n\2,v=0;while(x<=n&&v==0, r=n-x; if(isprime(r), v=1); x=2*x);e=1-v,e=0);e}

for(n=1,10000,if(espolignac(n)&&bigomega(n)==2,write1("final.txt",n,", ")))
```

Así quedaría el listado hasta 10000:

905, 959, 1199, 1207, 1211, 1243, 1271, 1477, 1529, 1541, 1589, 1649, 1807, 1829, 1927, 1969, 1985, 2171, 2231, 2263, 2279, 2429, 2669, 2983, 2993, 3029, 3149, 3215, 3239, 3341, 3353, 3431, 3505, 3665, 3817, 3845, 3985, 4063, 4151, 4195, 4573, 4589, 4633, 4717, 4781, 4811, 4841, 4843, 4855, 5143, 5609, 5617, 5729, 5731, 5755, 5761, 5771, 5917, 5951, 6001, 6065, 6119, 6161, 6193, 6283, 6403, 6433, 6463, 6509, 6535, 6539, 6731, 6757, 6821, 6941, 7169, 7199, 7289, 7319, 7343, 7379, 7387, 7405, 7431, 7747, 7783, 7799, 7807, 7811, 7813, 7913, 7961, 8023, 8031, 8141, 8159, 8257, 8399, 8411, 8587, 8621, 8873, 8915, 8921, 8981, 9101, 9115, 9307, 9517, 9557, 9569, 9641, 9809, 9959, Como curiosidad, ninguno de los primeros números del listado es múltiplo de 3. Hay que esperar a llegar a 7431 y 8031 para que aparezca.

De igual forma se pueden buscar otros tipos.

Cuadrados:

1
40401
62001
96721
121801
192721
326041
410881
555025
660969
683929
772641
786769
822649

Triangulares

1
46971
79003
93961
166753
203203
224785
286903
334153

Entre la sucesión de Fibonacci solo hemos encontrado dos (con cota 100000), el 1 y el 1597.

Como no se advierte ninguna propiedad especial, lo dejamos por ahora.

NÚMEROS DE FORTUNE

A los números que vamos a estudiar se les suele llamar afortunados, pero esa denominación puede confundirse con otras parecidas, como "números felices" o "de la suerte". Por ello los nombraremos según el primer matemático que los estudió, que fue Reo Franklin Fortune.

Para definirlos bien podemos comenzar recordando los números de Euclides. Son aquellos formados por el producto de los primeros números primos con el añadido de una unidad:

$E(n)=p_1*p_2*p_3*\ldots*p_n+1$
(Ver
https://es.wikipedia.org/wiki/N%C3%BAmero_de_Euclides
y
http://oeis.org/A006862)

Los conocimos en la demostración clásica de la infinitud de números primos, y unos son primos y otros compuestos, como 30031=59*509.

Al primer sumando en la definición se le llama *primorial,* y ya lo hemos estudiado en nuestro blog

http://hojaynumeros.blogspot.com/2012/02/el-primorial.html

El primorial se suele representar como N#, siendo N el número de factores primos consecutivos de su producto. Por ejemplo, 4#=2*3*5*7=210.

Llamemos Q(n) al primer primo posterior al número de Euclides E(n) de orden **n**, es decir, posterior a n#+1.
Puede ocurrir que la diferencia P(n) = Q(n)-n# sea un número primo, y en ese caso diremos que P(n) es **un número afortunado o de Fortune**. Este autor conjeturó que todos ellos serían primos. Es una cuestión no demostrada aún.

Por ejemplo, 2*3*5=30 es el tercer primorial, por lo que 31 es un número de Euclides. Su primo más próximo en orden creciente es 37, y la diferencia 37-30=7 es prima, luego 7 es un número afortunado.

En siguiente página de MathWorld puedes consultar lo más importante sobre estos números.
http://mathworld.wolfram.com/FortunatePrime.html

Búsquedas

Podemos reproducir la lista de números afortunados según el orden creciente de primoriales. El inconveniente, nada grave con una hoja de cálculo, es que resultarán desordenados y duplicados, pues existen soluciones iguales para distintos órdenes. Probamos con este tipo de búsqueda. Usaremos la siguiente función:

Function fortune(n)
dim i,k,p,q,j

k=0
j=2
p=1
for i=1 to n
p=p*j ‘Los primoriales se van formando en la variable ***p***
j=primprox(j) ‘Se añade un nuevo primo
next i
q=primprox(p+1)-p ‘Se restan el siguiente primo y el primorial
if esprimo(q) then k=q ‘Si la diferencia es prima, ***q*** es afortunado***.***
fortune=k
end function

La función devuelve un cero si el primo buscado no es afortunado o un número primo si lo es. Con ella podemos descubrir los primeros números de Fortune. Solo podemos llegar al orden 9 porque se produce desbordamiento:

Orden	Núm. Fortune
1	3
2	5
3	7
4	13
5	23
6	17
7	19
8	23
9	37

Están publicados en http://oeis.org/A005235 de forma no ordenada y con duplicados: 3, 5, 7, 13, 23, 17, 19, 23, 37, 61, 67, 61, 71, 47, 107, 59, 61, 109, 89, 103, 79, 151, 197, 101, 103,…

Para retardar el desbordamiento podemos usar la versión en PARI:

fortune(n)=my(k=0,j=2,p=1);for(i=1,n,p=p*j;j=nextprime(j));q=nextprime(p+1)-q;if(isprime(q),k=q);k

print(fortune(4))

Este sería el resultado:

```
1, 3
2, 5
3, 7
4, 13
5, 23
6, 17
7, 19
8, 23
9, 37
10, 61
11, 67
12, 61
13, 71
14, 47
15, 107
16, 59
17, 61
18, 109
19, 89
20, 103
? _
```

No parece que el tema dé para más con las herramientas de cálculo que usamos. Si consultas el tema en otras páginas descubrirás que es una cuestión limitada.

CADENAS DE CUNNINGHAM.

Este estudio se dedica a explicar los procedimientos para estudiar las cadenas de Cunningham con ayuda de la hoja de cálculo y del lenguaje PARI. Su definición y propiedades las puedes consultar en

https://es.m.wikipedia.org/wiki/Cadena_de_Cunningham

Para comenzar nuestro trabajo, solo necesitamos conocer la generación de una cadena de este tipo:

- Elegimos un número primo cualquiera.
- Lo sometemos a la recurrencia $p_{i+1} = 2\,p_i + 1$ (cadena de Cunnigham de primera especie) o bien a la recurrencia $p_{i+1} = 2\,p_i - 1$ (cadena de Cunningham de segunda especie) .
- Interrumpimos la recurrencia cuando el resultado no sea primo.

Aquí solo estudiaremos las cadenas de primera especie, porque contienen las propiedades más interesantes.

Todos los elementos de una de estas cadenas serán primos de Sophie Germain salvo el último y todos serán primos *seguros* salvo el primero.

(Ver
https://es.m.wikipedia.org/wiki/N%C3%BAmero_primo_de_Sophie_Germain
https://en.m.wikipedia.org/wiki/Safe_prime)

Una cadena de este tipo se llama completa si no se puede prolongar más, tanto con términos mayores como menores. Esas son las que estudiaremos aquí.

Es fácil ver que el primer término no ha de ser "primo seguro", pues (p-1)/2 no pertenece a la cadena y no es primo. Igualmente, el último no puede ser de Sophie Germain, porque 2p+1 no será primo.

En primer lugar crearemos una función que nos devuelva la cadena de Cunningham que se puede generar a partir un número primo, aunque este no sea el origen de una cadena completa, por ser a su vez generado por otro primo anterior. La función posee este código:

Function cunningham$(p)
Dim s$
Dim m

s$ = "" 'Esta variable recibirá la cadena en modo texto
If esprimo(p) Then 'Solo se trabaja con primos
m = 2 * p + 1 'Primer paso en la iteración

While esprimo(m) ‘Mientras sea primo, se continua
s$ = s$ + Str$(m) ‘Se recogen los elementos de la cadena
m = 2 * m + 1
Wend
End If
cunningham = s
End Function

Con esta función es fácil ver si un número primo produce una cadena de al menos dos elementos. Los primeros resultados son:

Inicio	Cadena
2	5 11 23 47
3	7
5	11 23 47
11	23 47
23	47
29	59
41	83 167
53	107
83	167
89	179 359 719 1439 2879

En esta tabla observamos algún detalle interesante:
Los inicios 2, 3, 5, 11, 23,…, como era de esperar, son primos de Sophie Germain, en los que si p es primo, 2p+1 también lo es. Los tienes en

http://oeis.org/A005384.

Los finales de cadena son primos que no pertenecen a ese tipo, como el 7 y el 47. Vemos en la tabla cierres con 47, 107 o 167.

Las cadenas se solapan, porque 11 pertenece a una y genera otra. Para evitar esto, la condición de que el número inicial sea primo habrá de ser completada con la de que (p-1)/2 no lo sea (*And Not esprimo((p - 1) / 2)*). De esa forma crearemos cadenas completas, sin solapamientos. Añadimos esa condición a la función y obtenemos:

Inicio	Cadena
2	5 11 23 47
3	7
29	59
41	83 167
53	107
89	179 359 719 1439 2879
113	227
131	263
173	347
191	383
233	467
239	479
251	503
281	563
293	587
419	839

Ahora ya sí tenemos cadenas completas, en las que según la teoría, los inicios son primos de Sophie Germain pero no primos seguros, que es la condición que habrás leído en la teoría.

Estos inicios de cadenas los tienes en http://oeis.org/A059453.

Llama la atención que muchas cadenas solo tienen dos elementos. Para estudiar sus longitudes bastará cambiar el código de la función, de forma que devuelva esa variable. Podemos asignar un 0 a los números que no son posibles inicios y después, con el mismo algoritmo, devolver las longitudes de las cadenas en lugar de su contenido.

El nuevo código sería:

```
Function lcunningham$(p)
Dim s
Dim m

If esprimo(p) And Not esprimo((p - 1) / 2) Then
s = 1
m = 2 * p + 1
While esprimo(m)
s = s + 1
m = 2 * m + 1
Wend
End If
lcunningham = s
End Function
```

Ahora la variable ***s*** cuenta los elementos en lugar de incorporarlos al texto. La nueva tabla sería esta:

Inicio	Longitud
2	5
3	2
29	2
41	3
53	2
89	6
113	2
131	2
173	2
191	2
233	2
239	2
251	2
281	2
293	2

Para estadísticas y clasificaciones es más útil que la que devuelve un texto. La podemos traducir a PARI.

lc(p)=my(c=0,m=2*p+1);if(p==2,c=5,if(isprime(p)&&!isprime((p-1)/2),c=1;while(isprime(m),c+=1;m=2*m+1)));c

Aquí asignamos un 5 al valor 2, para sacarlo del algoritmo general, y el resto es traducción del lenguaje de Excel.

Los inicios de la tabla anterior se pueden reproducir con ***forprime(n=2,500,if(lc(n)>=2,print1(n,", ")))***

Así resultan los primeros inicios:
2, 3, 29, 41, 53, 89, 113, 131, 173, 191, 233, 239, 251, 281, 293, 419, 431, 443, 491,…

Sustituyendo lc(n)>=2 por otras condiciones, como lc(n)==2, lc(n)==3, lc(n)<5…podemos clasificar las cadenas según su longitud. Esto ya está estudiado, por lo que nos limitaremos a comprobar algunos resultados:

lc(n)==3

Nos resultan las cadenas de longitud 3, con inicios
41, 1031, 1451, 1481, 1511, 1811, 1889, 1901, 1931, 3449, 3491, 3821, 3911,…

Están recogidos en http://oeis.org/A059762.

lc(n)==5

2, 53639, 53849, 61409, 66749, 143609, 167729, 186149, 206369, 268049, 296099, 340919, 422069, 446609,

Observamos que entre 2 y 50000 no hay soluciones. Puedes estudiar estos números en

http://oeis.org/A059764.

Si cambiamos la condición, es posible que los resultados no estén publicados.

lc(n)>=4

2, 89, 509, 1229, 1409, 2699, 3539, 6449, 10589, 11549, 11909, 12119, 17159, 19709, 19889, 22349, 26189, 27479, 30389, 43649,…

En efecto, al menos en OEIS no se recoge esta sucesión. No seguimos, porque se reduciría a una casuística.

Parece ser que se ha llegado a encontrar cadenas de 13 elementos (en el momento de escribir esto probablemente se habrá sobrepasado este número).
Con nuestras herramientas podemos encontrar los primeros números que son inicios de cadenas de longitud ocho. Los primeros son 19099919 y 52554569.

Con la función *cunningham* podemos encontrar esas cadenas:

19099919, 38199839, 76399679, 152799359, 305598719, 611197439, 1222394879, 2444789759.
52554569, 105109139, 210218279, 420436559, 840873119, 1681746239, 3363492479, 6726984959.

También hemos encontrado la primera cadena con nueve elementos:

85864769, 171729539, 343459079, 686918159, 1373836319, 2747672639, 5495345279, 10990690559, 21981381119.

Con hoja de cálculo no llegaremos más allá.

Estadísticas

Es curioso que el número de cadenas por intervalos se mantiene casi constante. En la siguiente tabla hemos estudiado intervalos de 5000 números, tomando nota del número de cadenas y el promedio de sus longitudes:

Total de cadenas y longitud media

De 1 a 5000	597	1,20771
De 5001 a 10000	517	1,15474
De 10001 a 15000	481	1,18087
15001 a 20000	477	1,15304
20001 a 25000	462	1,13853
25001 a 30000	446	1,06867
30001 a 35000	458	1,12664
35001 a 40000	439	1,14806
40001 a 45000	433	1,13857
45001 a 50000	432	1,12269

Observamos una gran semejanza en los datos, con ligera tendencia a disminuir.
Se observa la misma tendencia si los intervalos tienen longitud 50000:

De 50000 en 50000		
De 2 a 50000	4742	1,15099
De 50001 a 100000	4180	1,13254
De 100001 a 150000	4005	1,12784
150001 a 200000	3886	1,11863

Se deja como propuesta comparar estos datos con las distribuciones de números primos y de los de Sophie Germain.

MULTIPOLIGONAL

Recordemos que los números poligonales son aquellos que pueden organizar sus unidades en forma gráfica de polígono

(https://es.wikipedia.org/wiki/N%C3%BAmero_poligonal).

Llamamos **dimensión k** a su número de lados y **orden n** al número de unidades por lado. Su expresión algebraica es

$$P_{n.k} = \frac{n(n(k-2)-(k-4))}{2}$$

Con nuestra calculadora de números figurados puedes calcularlos fácilmente.

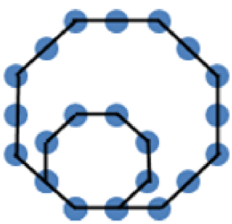
(Descarga libre desde

http://www.hojamat.es/sindecimales/aritmetica/herramientas/hoja/calcupol.xlsm)

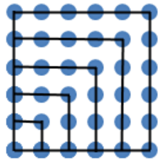
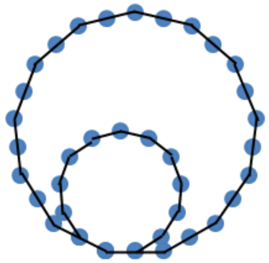
Por ejemplo, el número octogonal de orden 3 será igual a 21, ya que

$P_{8,3}$ = 3*(3*6-4)/2 = 3*14/2 = 21

Gráficamente:



El problema que abordaremos ahora es el de clasificar aquellos números poligonales que lo son tomando distintos órdenes y dimensiones. Por ejemplo, el número 36 se puede representar con cuatro lados y también con trece:

En efecto:

$P_{4,6}$ = 6*(6*2-0)/2 = 6*6 = 36

P13,3 = 3*(3*11-9)/2 =3*24/2 = 36

¿De cuántas formas poligonales se puede expresar un número? Es evidente que todos ellos son poligonales de orden 2, incluso los primos, ya que es posible ordenar las unidades sobre un perímetro de **n** lados. Buscamos ahora aquellos que admitan varias representaciones.

Comenzaremos con una búsqueda ordenada. Nuestra costumbre es usar una función en Basic de Excel o Calc. Podría ser esta:

```
Function mpolig$(n)
Dim i
Dim s$

If n < 3 Then mpolig = "NO": Exit Function
s$ = ""
For i = 3 To n
If espoligonal(n, i) Then s$ = s$ + " #" + Str$(i) + ", "
+ Str$(quepoligonal(n, i))
Next i
mpolig = s
End Function
```

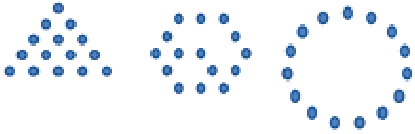
Esta función es de tipo texto, para poder añadir todos los tipos de poligonal que admite un número. Comienza desechando los números inferiores a 3. Después aplica un criterio, *espoligonal*, para ver si es poligonal o no. Si lo es, aplica la función *quepoligonal*, que encuentra el orden y dimensión. Si todo va bien, añade el nuevo polígono a la serie.

Estas dos funciones, *espoligonal* y *quepoligonal*, son algo teóricas, por lo que las añadimos a un Apéndice.

Por ejemplo, si aplicamos esta función al número 15, resultan tres formas de ser poligonal, y se presentan de esta forma:

MPOLIG(15)= # 3, 5 # 6, 3 # 15, 2

Significa que 15 puede ser un número triangular de lado 5, o bien hexagonal de lado 3, o, por último, un polígono de 15 lados. A partir de ahora no consideraremos esos polígonos formados por tantos lados como indique el número. Gráficamente:



Es fácil ver que las tres estructuras están compuestas por 15 unidades.

El primer número que presenta cuatro modalidades de poligonal es el 36:

MPOLIG(36)= # 3, 8 # 4, 6 # 13, 3 # 36, 2

Puede presentar 3, 4, 13 y 36 lados.

A estos números les podemos denominar *multipoligonales*. En el extremo opuesto figuran aquellos números que solo admiten forma de polígono simple, con lados de una unidad. Entre ellos están los primos y otros que no lo son, como el 38, que no admite ninguna otra forma poligonal salvo la trivial de un polígono de 38 lados. Este tipo de números está recogido en http://oeis.org/A090467.

Es fácil transformar la función *mpolig* en otra que simplemente cuente las soluciones de los distintos tipos de poligonal que admite un número. Quedaría así:

```
Function npolig(n)
Dim i, p

If n < 3 Then npolig = 0: Exit Function
p = 0
For i = 3 To n
If espoligonal(n, i) Then p = p + 1
Next i
npolig = p
End Function
```

Su funcionamiento es fácil de entender. Por ejemplo, con ella podemos listar los números que admiten cuatro representaciones como poligonales. Los primeros son:

36		4	# 3, 8 # 4, 6 # 13, 3 # 36, 2
45		4	# 3, 9 # 6, 5 # 16, 3 # 45, 2
66		4	# 3, 11 # 6, 6 # 23, 3 # 66, 2
81		4	# 4, 9 # 7, 6 # 28, 3 # 81, 2
105		4	# 3, 14 # 12, 5 # 36, 3 # 105, 2
120		4	# 3, 15 # 6, 8 # 41, 3 # 120, 2
153		4	# 3, 17 # 6, 9 # 52, 3 # 153, 2
171		4	# 3, 18 # 13, 6 # 58, 3 # 171, 2
190		4	# 3, 19 # 6, 10 # 33, 4 # 190, 2
196		4	# 4, 14 # 11, 7 # 34, 4 # 196, 2
210		4	# 3, 20 # 5, 12 # 71, 3 # 210, 2
261		4	# 9, 9 # 19, 6 # 88, 3 # 261, 2
280		4	# 8, 10 # 15, 7 # 48, 4 # 280, 2
351		4	# 3, 26 # 25, 6 # 118, 3 # 351, 2
378		4	# 3, 27 # 6, 14 # 127, 3 # 378, 2
396		4	# 9, 11 # 28, 6 # 133, 3 # 396, 2
400		4	# 4, 20 # 16, 8 # 68, 4 # 400, 2

En la primera columna figuran los números y en la segunda el número 4 y los distintos tipos de poligonal que admiten.

Los listados de los números con un número dado de tipos de poligonal están todos publicados en sus casos más sencillos. Por ejemplo, los anteriores figuran en http://oeis.org/A195528

Números con dos tipos determinados de poligonal

Para saber si un número es de dos tipos determinados basta usar esta función:

Function doblepolig(n, p, q) As Boolean

If espoligonal(n, p) And espoligonal(n, q) Then doblepolig = True Else doblepolig = False

End Function

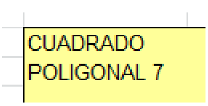
Es sencilla de entender, ya que exige que sea poligonal de tipo **p** y también de tipo **q**, que son los parámetros que acompañan a **n**.

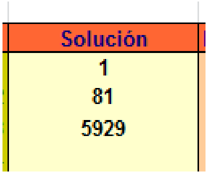
No vamos a recorrer todos los casos, ya que los valores del número de lados aumentan muy pronto, y se hacen inabordables.

Por ejemplo, con hoja de cálculo solo podemos obtener de forma razonable dos números que son heptagonales y cuadrados, el 81 y el 5929 (además del trivial 1). El siguiente, 2307361, necesita más tiempo de cálculo, y no digamos los que siguen: 12328771225, 4797839017609, 350709705290025, 25635978392186449, 9976444135331412025,…

(Ver http://oeis.org/A036354)

Con nuestro Buscador de Naturales (http://www.hojamat.es/sindecimales/divisibilidad/herramientas/herrdiv.htm#buscador) es fácil plantear esta cuestión, aunque tampoco llegamos muy lejos:





Solución
1
81
5929

En otros casos no se puede aplicar este método porque en el Buscador no podemos repetir la palabra POLIGONAL.

Un caso concreto

(Complemento con uso de técnicas algo más avanzadas)

¿Qué números son hexagonales y también cuadrados?

Si los buscamos con la función *doblepolig* y parámetros 4 y 6, nos resulta el número 1225, pero el siguiente 1413721, ya necesita unos minutos de proceso. Intentamos una aproximación algebraica:

Un número cuadrado de orden n tiene como fórmula $\mathbf{n^2}$, y un hexagonal de orden m, **m(2m-1)**. Igualamos:

Sería $n^2=2m^2-m$

Multiplico por 2:

$2n^2=4m^2-2m$

Completo un binomio al cuadrado:

$2n^2=(2m)^2-2*2m*1/2+(1/2)^2-(1/2)^2$

$2n^2=(2m-1/2)^2-(1/2)^2$

Multiplico por 4

$8n^2=(4m-1)^2-1$

$(4m-1)^2-8n^2=1$

Llamando x=8m-1 e y=n, logramos el planteo de una ecuación de Pell: $x^2-8y^2=1$

(Ver mi entrada

http://hojaynumeros.blogspot.com/2010/02/ecuacion-de-pell.html)

Acudimos a nuestra herramienta

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

Rellenamos los parámetros 8, RAIZ(8) y 1 y obtenemos las soluciones:

X	Y	
3	**1**	**+1 ó -1**
17	6	1
99	35	1
577	204	1
3363	1189	1
19601	6930	1
114243	40391	1
665857	235416	1

Al pasar de **X** a **n**, deberemos desechar las soluciones no enteras. Aplicamos a **X** la función (X+1)/8 para volver a **n**:

3	1	+1 ó -1		M
17	6	1		4,5
99	35	1		25
577	204	1		144,5
3363	1189	1		841
19601	6930	1		4900,5
114243	40391	1		28561
665857	235416	1		166464,5

Nos quedamos con las soluciones enteras. La primera es Y=N=35 y X=25. Vemos que, efectivamente, se cumple que el cuadrado de 35, 1225, coincide con el hexagonal de 25, $2*25^2-25=1250-25=1225$.

Veamos la segunda solución entera:

Y=N=1189, X=841, con lo que $1189^2=1413721=2*841^2-841$

De igual forma obtendríamos la tercera, $40391^2=1631432881$

(Puedes comprobarlo en http://oeis.org/A046177)

Recurrencia

Esta parte la desarrollaremos sin justificar.

En una ecuación de Pell de parámetro 8, las soluciones se obtienen a partir de las primeras X=3, Y=1 mediante las recurrencias

$X_{n+1}=3X_n+8Y_n$ y $Y_{n+1}=X_n+3Y_n$

Puedes comprobarlo en la tabla de más arriba.

Como las soluciones enteras aparecen cada dos pasos, las recurrencias quedarán

$X_{n+2}=3(3X_n+8Y_n)+8(X_n+3Y_n)=17X_n+48Y_n$

$Y_{n+2}=3X_n+8Y_n+3*(X_n+3Y_n)=6X_n+17Y_n$

Así, de x=99, y=35, obtenemos:

X=17*99+48*35=3363

Y=6*99+17*35=1189

Estas serían las siguientes soluciones. Reiterando, obtendríamos todas las demás:

X=17*3363+48*1189=114243

Y=6*3363+17*1189=40391

Este ha sido un ejemplo concreto, sin demasiada dificultad algebraica. Las demás coincidencias entre dos tipos se resuelven de forma similar, pero quizás con un desarrollo más complejo.

Apéndice

Función espoligonal

Devuelve VERDADERO o FALSO

```
Function espoligonal(n, k) As Boolean
Dim d
Dim e As Boolean

e = False
```

```
d = (k - 4) ^ 2 + 8 * n * (k - 2)
If escuad(d) Then
If esentero((k - 4 + Sqr(d)) / 2 / (k - 2)) Then e = True
End If
espoligonal = e
End Function
```

Función quepoligonal

Encuentra la dimensión de un número como poligonal

```
Function quepoligonal(n, k)
Dim d
d = Sqr((k - 4) ^ 2 + 8 * n * (k - 2))
If d <> Int(d) Then quepoligonal = 0 Else quepoligonal = (k - 4 + d) / (2 * k - 4)
End Function
```

NÚMEROS DE ULAM

Se llaman *números de Ulam* a los que forman una sucesión construida de la siguiente forma:

Se declara u(1)=1 y u(2)=2 (veremos que esto se puede alterar) y después definiremos u(n+1) como el primer número que se pueda expresar como **suma de dos números de Ulam anteriores distintos, de forma única**.

Los creó el matemático polaco Stanislaw Ulam y los publicó en SIAM Review en 1964.

Puedes ampliar este concepto en las páginas

https://es.wikipedia.org/wiki/N%C3%BAmeros_de_Ulam

http://www.grupoalquerque.es/mate_cerca/paneles_2014/221_Numeros%20de%20Ulam.pdf

https://www.gaussianos.com/feliz-ano-2012/

El 1 y el 2 se toman de forma arbitraria. El siguiente deberá ser el 3, ya que 3=1+2 de forma única. También está claro que el cuarto debe ser 4=1+3, pues 4=2+2 no sería válido por ser iguales los sumandos.

Deberíamos rechazar el 5, pues 5=1+4=2+3. El 6 sí nos vale, pues 6=2+4, siendo no válida la suma 6=3+3.

Por tanto, los primeros términos de la sucesión de Ulam serán 1, 2, 3, 4, 6. Es sencillo buscar los siguientes: 8, 11, 13, 16, 18, 26,…

Tienes más elementos en http://oeis.org/A002858, junto con una gran cantidad de comentarios sobre estos números. Aquí nos interesarán aspectos algorítmicos. Veamos alguno:

Encontrar el número de Ulam de orden N

Resolveremos la cuestión a través de una función que nos devuelva el término enésimo de la sucesión. Esto tiene el inconveniente de que hay que ir tomando nota de los términos anteriores. Los lenguajes avanzados lo resuelven mediante **una lista**, tal como veremos en PARI. En hoja de cálculo se pueden construir fácilmente listas mediante las columnas, usando una variable que recuerde el número de términos de la lista y unos procedimientos para escribir y leer en ella. No es complicado. Ya lo usamos en otra sucesión, la de Mian-Chowla:

https://hojaynumeros.blogspot.com/2016/03/sucesion-de-mian-chowla.html

Aquí lo implementaremos de forma más simple, dimensionando un vector con 2000 componentes. El inconveniente es que no podremos pasar de esa cantidad, salvo que modifiquemos la dimensión, pero resulta más manejable.

Una vez determinemos la lista, en este caso u(2000), rellenaremos en primer lugar los elementos u(1)=1 y u(2)=2. Después habrá que ir probando los siguientes números hasta ver si proceden de una suma única de elementos distintos o no. Crearemos una variable **m** que cuente las sumas posibles, y si vale 1, incorporamos un nuevo elemento a la lista. Al llegar a **n** elementos, paramos el cálculo y devolvemos ese resultado. El código de la función para Excel y Calc es el siguiente, en el que hemos añadido los parámetros **a** y **b** por si deseamos iniciar la sucesión en otro par de números que no sean 1, 2:

```
Public Function ulam(a, b, n)
Dim u(2000) ‘Usamos un vector o matriz, pero puede ser una lista
Dim i, j, k, m, uu
Dim noes As Boolean

u(1) = a: u(2) = b ‘Se rellenan los primeros términos
If n = 1 Or n = 2 Then ulam = u(n): Exit Function ‘Primeros dos casos
For i = 3 To n
noes = True
uu = u(i - 1) + 1 ‘La variable uu recorre los números de Ulam previos
While noes
m = 0
For j = 1 To i - 1
For k = j + 1 To i - 1
If j <> k And u(j) + u(k) = uu Then m = m + 1 ‘Cuenta las sumas distintas
Next k
Next j ‘A continuación actúa cuando la suma es única
If m = 1 Then u(i) = uu: noes = False
Else uu = uu + 1
Wend
Next i
ulam = uu
End Function
```

N	Ulam(N)
1	1
2	2
3	3
4	4
5	6
6	8
7	11
8	13
9	16
10	18
11	26
12	28
13	36
14	38
15	47
16	48
17	53
18	57
19	62
20	69

En la siguiente tabla, mediante esta función, hemos representado los 20 primeros números de Ulam:

Recuerda que solo podremos actuar sobre los 2000 primeros, tal como hemos definido la función. Este inconveniente no existe si usas lista en el lenguaje PARI. En el siguiente listado observarás que se comienza creando la lista **v**: “***my(v=List(),”*** y, después, para incorporar un elemento a la lista se usa “**lisput**”. Lo que sigue es difícil de leer, pero se puede comprobar que contiene las mismas ideas que en la función de Excel, con alguna ligera variante. Esta es la función propuesta:

ulam(n)=my(v=List(),i,j,k,o,uu,m);listput(v,1);listput(v,2);for(i=3,n,o=1;uu=v[i-1]+1;while(o==1,m=0;for(j=1,i-1,for(k=j+1,i-1,if(v[j]+v[k]==uu&&j<>k,m+=1)));uu+=1;if(m==1,uu=uu-1;listput(v,uu);o=0)));uu

Por no complicar más (ya es bastante oscuro), no hemos implementado la función para n=1 o n=2, por lo que el listado general comenzará en el 3:

```
3, 4, 6, 8, 11, 13, 16, 18, 26, 28, 36, 38, 47, 48, 53, 57, 62, 69, 72, 77, 82,
87, 97, 99, 102, 106, 114, 126, 131, 138, 145, 148, 155, 175, 177, 180, 182, 189
, 197, 206, 209, 219, 221, 236, 238, 241, 243, 253,
?
```

Así que con esta función podemos crear la lista de los números de Ulam, pero nos puede interesar también si un número es de Ulam o no. Por ejemplo, ¿Qué número de Ulam sigue al 200?

Para ello disponemos de otra función basada en la anterior. Se podían refundir ambas en una sola, pero no compensaba el esfuerzo. La orientación que se propone aquí es más lenta, pero más fácil de entender.

Ver si un número es de Ulam

Llamaremos ***esulam*** a una función que nos devuelva un 0 si el número propuesto no pertenece a la sucesión de Ulam y que nos dé su número de orden en caso de que pertenezca.

Public Function esulam(a, b, n)
Dim i, uu, k

If n = a Then esulam = 1: Exit Function ‘Caso u(1)
If n = b Then esulam = 2: Exit Function ’Caso u(2)
i = 3: k = 0: uu = 0
While uu <= n ‘Busca elementos menores que ***n***
uu = ulam(a, b, i)
If uu = n Then k = i ‘Si se llega a ***n***, es que pertenece, y se toma nota del orden ***k***. Si no, k=0
i = i + 1
Wend

esulam = k
End Function

Con esta función averiguamos que el primer número de Ulam posterior al 200 es el 206:

N	ESULAM(N)
200	0
201	0
202	0
203	0
204	0
205	0
206	42
207	0
208	0

Los primeros números de Ulam que son números primos son:2, 3, 11, 13, 47, 53, 97, 131, 197, 241, 409, 431, 607, 673, 739, 751, 983, 991, 1103, 1433, 1489.

Como ejercicio, puedes comprobar los siguientes listados de números de Ulam que cumplen otras condiciones:

Cuadrados: 1, 4, 16, 36, 324, 400, 441,…

Triangulares: 1, 3, 6, 28, 36, 253,…

Capicúas (contando los de una cifra): 1, 2, 3, 4, 6, 8, 11, 77, 99, 131, 282, 363, 414, 434, 585, 646,…

Otras sucesiones de Ulam

Podemos cambiar los valores iniciales 1 y 2 por otros (por eso se incluyeron los parámetros a y b en nuestras funciones). A continuación se copian algunas, con indicación de su número en OEIS:

(a,b)	Dirección	Sucesión
(1, 2)	A002858	1, 2, 3, 4, 6, 8, 11, 13, 16, 18, ...
(1, 3)	A002859	1, 3, 4, 5, 6, 8, 10, 12, 17, 21, ...
(1, 4)	A003666	1, 4, 5, 6, 7, 8, 10, 16, 18, 19, ...
(15)	A003667	1, 5, 6, 7, 8, 9, 10, 12, 20, 22, ...
(2, 3)	A001857	2, 3, 5, 7, 8, 9, 13, 14, 18, 19, ...
(2, 4)	A048951	2, 4, 6, 8, 12, 16, 22, 26, 32, 36, ...
(2, 5)	A007300	2, 5, 7, 9, 11, 12, 13, 15, 19, 23, ...

Con esto terminamos. No parece que este estudio dé para más.

NÚMEROS DUFFINIANOS

Estos números, llamados así por Richard Duffy, son números compuestos que son primos con la suma de sus divisores, es decir, con el valor de la función SIGMA

(σ). En ellos no existe ningún divisor común entre N y σ(N).

Por ejemplo, es duffiniano el 111, que es compuesto, ya que 111=3*37, y la suma de sus divisores es σ(111)=111+37+3+1=152, cuya descomposición factorial es 2^3*19. Los factores primos de 111 son 3 y 37, mientras que los de la suma de sus divisores son 2 y 19, luego son primos entre sí y 111 es duffiniano.

Se excluyen los primos porque cumplen la condición de forma trivial: si p es primo, σ(p)=1+p, y dos números consecutivos siempre son primos entre sí (intenta calcularles el M.C.D.).

En el resto del texto podremos aplicar la propiedad de que SIGMA es una función multiplicativa (ver http://www.hojamat.es/sindecimales/divisibilidad/teoria/teordivi.pdf), es decir, que si **m** y **n** son primos entre sí, se cumple que **σ(a.b)=σ(a).σ(b)**

Lista de los primeros números duffinianos

En este estudio no es necesario acudir a un algoritmo. Basta exigir que MCD(N,SIGMA(N))=1.

En Excel disponemos de la función ***M.C.D*** y en PARI ***gcd***. Respecto a la función SIGMA, no está implementada en hojas de cálculo, pero puedes usar la diseñada en

https://hojaynumeros.blogspot.com/2019/10/la-funcion-sigma-y-sus-traslados.html

En PARI se usa la función ***sigma***, tal cual.

Puedes construir una lista en la que todos los números sean coprimos con los valores de SIGMA en ellos. Los primeros que obtendrás son los siguientes:

4, 8, 9, 16, 21, 25, 27, 32, 35, 36, 39, 49, 50, 55, 57, 63, 64, 65, 75, 77, 81, 85, 93, 98, 100, 111, 115, 119, 121, 125, 128, 129, 133, 143, 144, 155, 161, 169, 171, 175, 183, 185, 187, 189, 201, 203, 205, 209, 215, 217, 219, 221, 225, 235, 237, 242, 243, 245, 247,…

Esos son los primeros números duffinianos. Los tienes publicados en http://oeis.org/A003624

Como su búsqueda no presenta problemas, nos dedicaremos aquí a estudiar tipos especiales y a explicar propiedades.

Tipos especiales

Hemos visto que entre ellos no hay números primos, pero sí observamos que pertenecen a la lista potencias de primos, como 8, 9, 16, 27,…

Para estudiarlos nos basta con el siguiente desarrollo:

$$\sigma(p^r) = 1 + p + p^2 + p^3 + \cdots + p^r$$

Vemos que la potencia únicamente es divisible entre p y sus primeras potencias, pero en el segundo siempre obtendríamos un resto igual a 1 al dividir. Por tanto:

Todas las potencias de un número primo y exponente mayor que 1 son números duffinianos.

Esta propiedad garantiza la infinitud de los números de este tipo.

Semiprimos

Los número semiprimos N se descomponen como N=p*q, siendo p y q primos. Si ambos son iguales, N será el cuadrado de un número primo, y acabamos de ver que sí será duffiniano. Si son distintos p y q, no todos estos semiprimos lo serán. En efecto, por la propiedad multiplicativa, si N=pq con p<>q, tendremos

$\sigma(N)= \sigma(p)* \sigma(q)=(1+p)(1+q)$

Para que N sea duffiniano, ha de ser primo con $\sigma(N)$, lo que exige que p sea primo con q+1 y q lo sea con p+1. No todos los pares de primos cumplen esta condición.

Los primeros que sí la cumplen son:

21, 35, 39, 55, 57, 65, 77, 85, 93, 111, 115, 119, 129, 133, 143, 155, 161, 183, 185,…

Por ejemplo, 35=5*7, $\sigma(35)=(5+1)(7+1)=48$, que es primo con 35.
Por el contrario, existen otros semiprimos que, o bien **p** tiene factores comunes con **q+1** o **q** los posee con **p+1**. Los primeros son estos:

15, 33, 51, 69, 87, 91, 95, 123, 141, 145, 159, 177, 213, 249, 267, 287,…

Así, por ejemplo, 69=3*23 y $\sigma(69)=(3+1)(23+1)=96$, que posee un divisor común con 69, que es el 3. Esto se ha producido porque 3 no es primo con (23+1)

Resumiendo, existen semiprimos duffinianos y otros que no lo son, dependiendo de las relaciones entre sus dos factores.

Cuadrados

Todos los cuadrados de primos pertenecen a este tipo que estudiamos, pero también existen cuadrados de compuestos. En la tabla se han incluido los primeros, junto con el valor de SIGMA y la descomposición factorial de ambos, para comprobar que no presentan factores comunes.

Cuadrado	Sigma	Factores	Factores de Sigma
16	31	[2,4]	[31,1]
36	91	[2,2][3,2]	[7,1][13,1]
64	127	[2,6]	[127,1]
81	121	[3,4]	[11,2]
100	217	[2,2][5,2]	[7,1][31,1]
144	403	[2,4][3,2]	[13,1][31,1]
225	403	[3,2][5,2]	[13,1][31,1]
256	511	[2,8]	[7,1][73,1]
324	847	[2,2][3,4]	[7,1][11,2]
400	961	[2,4][5,2]	[31,2]
484	931	[2,2][11,2]	[7,2][19,1]
576	1651	[2,6][3,2]	[13,1][127,1]
625	781	[5,4]	[11,1][71,1]
676	1281	[2,2][13,2]	[3,1][7,1][61,1]
729	1093	[3,6]	[1093,1]
784	1767	[2,4][7,2]	[3,1][19,1][31,1]
900	2821	[2,2][3,2][5,2]	[7,1][13,1][31,1]
1024	2047	[2,10]	[23,1][89,1]
1089	1729	[3,2][11,2]	[7,1][13,1][19,1]

Observamos que entre ellos figuran potencias de primos, que ya sabemos que pertenecen. Entre los de base compuesta, vemos que los hay de dos factores primos distintos y también de tres, como el 900, por lo que no parece que haya limitación en este detalle.

Triangulares y oblongos

Un número triangular es del tipo m(m+1)/2. Los primeros triangulares duffinianos son:

3, 21, 36, 55, 171, 253, 325, 351, 595, 741, 903, 1081, 1225, 1711, 1953,…

Por ejemplo, 325 es triangular, porque 325=25*26/2, el valor de σ(325)=434, y no tienen divisores comunes, ya que $325=5^2*13$ y 434=2*7*31

Con los oblongos la situación es muy distinta. Sólo he encontrado seis ejemplos, que suben pronto a números de trece cifras. He seguido buscando, y no he encontrado más ejemplos empleando un tiempo razonable.

Son estos: 2, 2450, 2827442, 3262865762, 3765344262050, 4345204015540082

Por ejemplo, 3262865762 es oblongo, porque 3262865762=57121*57122. Su función SIGMA tiene el valor de 5324420103. Ambos números son primos entre sí.

$3262865762=2*13^4*239^2$ y 5324420103=3*19*3019*30941

No tienen factores comunes.

Cubos y cuartas potencias

En estos dos caso deberemos excluir las potencias de primos, que ya sabemos con seguridad que son duffinianos.

Cubos

Usaremos una condición triple, y es que sea un cubo, también duffiniano y, por último, que su base no sea prima. Con este condicionamiento sólo obtenemos estos casos entre 2 y 50000:

Cubo	Sigma	Base no prima
64	127	4
512	1023	8
729	1093	9
4096	8191	16
9261	16000	21
15625	19531	25
19683	29524	27
32768	65535	32
46656	138811	36

En la primera columna figuran los cubos obtenidos, que se ve que son primos con los valores de sigma de la segunda columna. En la tercera podemos observar que las bases pueden ser múltiplos de 2, 3, 5 o 7. No hay exclusiones.

Un ejemplo sería el 9261, que es el cubo de 21, por lo que sus factores primos son 3 y 7. La suma de divisores de 9261 es 16000, cuyos factores es claro que son 2 y 5. Por tanto, 9261 es duffiniano.

Cuartas potencias

Procedemos de la misma forma, y obtendremos estos primeros casos menores que 100000:

Potencia	Sigma	Base no prima
1296	3751	6
10000	24211	10
20736	61831	12
38416	86831	14
50625	94501	15

Así, 38416 es igual a 14^4, por lo que sus factores serán 2 y 7. Su función SIGMA tiene un valor de 86381, que es el producto de 31 por 2801, números primos que no coinciden con 2 y 7.

Duffinianos consecutivos

Estudiando la lista de duffinianos se observa que existen en ella consecutivos, como 8 y 9. No es difícil encontrar más ejemplos. En la lista siguiente figura el primer número del par. En ella también existen pares de consecutivos, que suponen un conjunto de tres:

8, 35, 49, 63, 64, 128, 143, 242, 323, 324, 391, 399, 484, 511, 512, 575, 578, 721, 722, 784, 799, 899, 900, 1024, 1057, 1156, 1250, 1295, 1351, 1443, 1444, 1681, 1921, 1936,…

En efecto, la lista descubre conjuntos de tres consecutivos {63, 64, 65}, {323, 324, 325}, {511, 512, 513},…

Con esto se puede dar por agotado el tema.

LA FUNCIÓN SIGMA Y SUS TRASLADOS

En este apartado investigaremos los números enteros positivos tales que al sumarles k unidades, el valor de su función sigma (suma de divisores) no cambia, es decir:

$$\sigma(n) = \sigma(n+k)$$

Existen muchos ejemplos según los valores de k, y recorreremos algunos para destacar sus propiedades.

Será de utilidad repasar la fórmula de esta función según la descomposición factorial del número. Es la siguiente:

$$\sigma(N) = \prod \frac{{p_i}^{e_i+1} - 1}{p_i - 1}$$

donde $\mathbf{p_i}$ son los factores primos de N y $\mathbf{e_i}$ sus exponentes. Cada factor también se puede interpretar como la suma de potencias del número primo correspondiente desde p^0 hasta p^e:

$$\sigma(N) = (1 + p_1 + {p_1}^2 + \cdots {p_1}^{e_1})\,(1 + p_2 + {p_2}^2 + \cdots {p_2}^{e_2})\ldots$$

Una implementación sencilla (para Excel o Calc) de esta función es la siguiente, escrita en código VBasic, aunque solemos usar otra basada en la descomposición factorial:

Public Function sigma(n)
Dim i, s
i = 1
s = n ‘La sigma se inicia con el valor de **n**
For i = 1 To n / 2 ‘El máximo divisor propio posible es n/2
If n / i = n \ i Then s = s + i ‘Si es divisor, se suma
Next i
sigma = s
End Function

Caso K=1

No es difícil construir un bucle de búsqueda de números consecutivos con la misma sigma. Los primeros son estos:

N	N+1	Sigma coincidente
14	15	24
206	207	312
957	958	1440
1334	1335	2160
1364	1365	2688
1634	1635	2640
2685	2686	4320
2974	2975	4464
4364	4365	7644
14841	14842	22932
18873	18874	28314
19358	19359	29040

Con el Buscador se puede reproducir fácilmente este listado. Basta plantear:

ES SUMDIV(N)=SUMDIV(N+1)
EVALUAR SUMDIV(N)

Resultado:

Solución	Detalles
14	24
206	312
957	1440
1334	2160
1364	2688
1634	2640
2685	4320
2974	4464
4364	7644

Están publicados en http://oeis.org/A002961

A002961 Numbers n such that n and n+1 have same sum of divisors.

14, 206, 957, 1334, 1364, 1634, 2685, 2974, 4364, 14841, 18873, 19358, 20145, 24957, 33998, 36566, 42818, 56564, 64665, 74918, 79826, 79833, 84134, 92685, 109214, 111506, 116937, 122073, 138237, 147454, 161001, 162602, 166934

En esta página se comenta que para valores de n < 2*10^10 el valor de sigma(n)/n está entre 1,5 y 2,25. No se sabe si esta sucesión es infinita.

Parece ser que 14 y 15 son los únicos semiprimos de la sucesión, y la sigma coincidente es 24 porque $\sigma(14)=\sigma(2*7)=(1+2)(1+7)=3*8=24$ y $\sigma(15)=\sigma(3*5)=(1+3)(1+5)=4*6=24$

Ni **p** ni **p+1** pueden ser primos en esta sucesión. Si **p** fuera primo, sería $\sigma(p)=1+p$, con lo que no podría alcanzar el valor de $\sigma(p+1)$. Si el que es primo es **p+1**, tendríamos $\sigma(p+1)=p+2$, con lo que los divisores propios de p deberían sumar 2, lo que no ocurre nunca.

206 y 207 son los siguientes (no semiprimos en este caso), porque

$\sigma(206)= \sigma(2*103)=(1+2)(1+103)=3*104=312$ y

$\sigma(207)= \sigma(3^2*23)=(1+3+9)(1+23)=13*24=312$

El código PARI que figura en la publicación citada no es el más compacto. Se puede usar preferiblemente este otro:

for(p=1,20000, if(sigma(p)==sigma(p+1), print1(p,", ")))

Hasta donde hemos explorado, la sigma común es múltiplo de 6.

Caso k=2

Aplicando la función sigma a dos números que se diferencien en dos unidades, resultan con resultados iguales los siguientes (primeras soluciones):

N	N+2	Sigma
33	35	48
54	56	120
284	286	504
366	368	744
834	836	1680
848	850	1674
918	920	2160
1240	1242	2880
1504	1506	3024
2910	2912	7056
2913	2915	3888
3304	3306	7200
4148	4150	7812
4187	4189	4320
6110	6112	12096
6902	6904	12960
7169	7171	7344

Con el Buscador:

Solución	Detalles
33	48
54	120
284	504
366	744
834	1680
848	1674
918	2160
1240	2880
1504	3024
2910	7056

Buscamos desde el número	1
Hasta el número	5000

Con estas propiedades:

ES SUMDIV(N)=SUMDIV(N+2)
EVALUAR SUMDIV(N)

Como en el caso anterior, era de esperar que estuviesen ya publicados:

A007373 Numbers n such that sigma(n+2) = sigma(n).

33, 54, 284, 366, 834, 848, 918, 1240, 1504, 2910, 2913, 3304, 4148, 4187, 6110, 6902, 7169, 7912, 9359, 10250, 10540, 12565, 15085, 17272, 17814, 19004, 19688, 21410, 21461, 24881, 25019, 26609, 28124, 30592, 30788, 31484, 38210, 38982, 39786, 40310, En esta sucesión, al igual que en la anterior, no hay primos, y parece que tampoco cuadrados. Las sigmas comunes que aparecen, también son múltiplos de 6 en este caso.

Sí figuran semiprimos en esta sucesión (las dos últimas columnas son los factores primos de las dos primeras. Al ser semiprimos, los exponentes de cada corchete son iguales a 1):

33	35	48	[3,1][11,1]	[5,1][7,1]
4187	4189	4320	[53,1][79,1]	[59,1][71,1]
7169	7171	7344	[67,1][107,1]	[71,1][101,1]
24881	24883	25200	[139,1][179,1]	[149,1][167,1]
25019	25021	25344	[127,1][197,1]	[131,1][191,1]
59987	59989	60480	[223,1][269,1]	[239,1][251,1]
77057	77059	77616	[251,1][307,1]	[263,1][293,1]

Por ejemplo, $\sigma(4187)=(1+53)(1+79)=4320$ y $\sigma(4189)=(1+59)(1+71)=4320$

Abreviamos. Para el siguiente caso tenemos:

CASO K=3

N	N+3	Sigma
382	385	576
8922	8925	17856
11935	11938	18432
31815	31818	63648
32442	32445	64896
61982	61985	98496

También están publicados.

A015861 Numbers n such that sigma(n) = sigma(n + 3).

382, 8922, 11935, 31815, 32442, 61982, 123795, 145915, 186615, 271215, 442362, 554715, 560382, 580635, 964535, 1191575, 1243375, 1369302, 1539942, 1642795, 2616702, 3141215, 3299062, 3556035, 3716895, 4201015, 5148294 (list; graph; refs; listen; history; text; internal format)

Por último el caso de diferencia 4:

No añadimos detalles.

N	N+4	Sigma
51	55	72
66	70	144
115	119	144
220	224	504
319	323	360
1003	1007	1080
2585	2589	3456
4024	4028	7560
4183	4187	4320
4195	4199	5040
5720	5724	15120
5826	5830	11664
5959	5963	6120
8004	8008	20160

A015863 Numbers n such that sigma(n) = sigma(n + 4).

51, 66, 115, 220, 319, 1003, 2585, 4024, 4183, 4195, 5720, 5826, 5959, 8004, 8374, 11659, 12367, 12561, 13581, 14338, 15365, 16116, 17840, 18718, 20541, 25130, 29393, 30170, 32665, 36516, 39913, 40660, *42423, 42922, 47841, 49762 (list; graph; refs; listen; history; text; internal format)*

Los casos k=5 y k=6 estàn también publicados. Dejamos esta primera cuestión.

Caso k=5 http://oeis.org/A015865

Caso k=6 http://oeis.org/A015866

Para investigar otros casos (por ejemplo, el 22) puedes construir un bucle (lo desarrollamos en VBasic) similar al siguiente:

For n=1 to 10000 ‘Hemos escrito 10000 como ejemplo
K=22 ‘El 22 también es un ejemplo
If sigma(n)=sigma(n+k) then msgbox(n) ‘Si coinciden las sigmas, lo presentamos
Next n

Ordenando la búsqueda nos ha resultado

N	N+22	Sigma coincidente
57	79	80
85	107	108
213	235	288
224	246	504
354	376	720
476	498	1008
568	590	1080
594	616	1440
812	834	1680
1218	1240	2880

Así puedes proceder en otros casos.

Diferencias que no se dan

Podemos investigar si desde **1** hasta **m** existen números que no cumplan la propiedad para un valor de k. Podría ser esta, que devuelve las diferencias que no se dan:

```
Public Function norepitesigma(m, k)
Dim i, s
Dim norepe As Boolean

i = 1: norepe = True: s = 0
While i <= m And norepe
If fsigma(i, 1) = fsigma(i + k, 1) Then norepe = False:
s = i
i = i + 1
Wend
norepitesigma = s
End Function
```

Para valores menores que 10000 estas son las primeras diferencias que no se dan (hay más):

1879
2111
2287
2520
2669
2700
2760
2820
2880
2894
2940
2978
3000
3054

Probamos con 100000 y las diferencias de la tabla anterior inferiores a 5000 desaparecen. Esto nos hace sospechar que, dada una diferencia entre números con sigmas iguales, se alcanza siempre un valor para el que es válida esa diferencia. Para verlo mejor podríamos invertir el punto de vista: dada una diferencia, averiguar en qué número se da. Este problema no tiene cota de búsqueda, por lo que la efectuaremos con cotas fijadas por nosotros. Podemos usar:

Public Function tienesigmacomun(n, c)
'Para una diferencia n, creamos un bucle con cota c hasta que aparezca esa diferencia
Dim k, s
Dim notiene

k = 1: notiene = True: s = 0
While notiene And k < c ‘Avanzamos si no aparece o llegamos a la cota
If fsigma(k, 1) = fsigma(n + k, 1) Then notiene = False: s = k

k = k + 1
Wend
tienesigmacomun = s
End Function

Con esta función podemos desechar las diferencias de la tabla anterior. Todas ellas aparecen con una cota de 100000:

Diferencia D	Aparece en K	K+D	sigma(K)	sigma(K+D)
1879	14390	16269	25920	25920
2111	13630	15741	25920	25920
2287	16172	18459	30576	30576
2520	12597	15117	20160	20160
2669	13618	16287	22320	22320
2700	10010	12710	24192	24192
2760	10332	13092	30576	30576
2820	10528	13348	24192	24192
2880	10556	13436	23520	23520
2894	12040	14934	31680	31680
2940	11781	14721	22464	22464
2978	10010	12988	24192	24192
3000	10395	13395	23040	23040
3054	10180	13234	21420	21420

Esto nos hace sospechar que todas las diferencias que planteemos terminarán por aparecer para algún valor.

Lo puedes investigar en PARI:

tiene(n)=local(c=100000,nr=1,k=1,s=0);while(nr==1&&k<c,if(sigma(k)==sigma(k+n),nr=0;s=k);k+=1);s

for(i=1,50000,if(tiene(i)==0,print(i))

Este código recorre desde 1 hasta 50000 para encontrar números que no puedan ser diferencias de sigmas con cota 100000. Descubre dos casos en los que no aparecen con esa cota 100000, que son 20160 y 22680, pero aparecen en los números 100776 y 113373 respectivamente. Esto nos hace sospechar que todos los números, buscando lo suficiente, podrán ser diferencias de otros números con sigmas equivalentes.

Cambiando los parámetros 1000000 y 5000 puedes intentar descubrir si alguna diferencia no aparece nunca para una cota de 1000000 o mayor. Con esta cuestión abierta terminamos el tema.

UNIDOS POR EL SOPF

Es tradición nuestra aprovechar cuestiones surgidas en Twitter. La de hoy se basa en una planteada por Juan Carlos Amez, @juankaamez, el día 28 de septiembre de 2019.

¿Podríamos probar que para cualquier valor k existe siempre al menos un valor n que verifica que: sopf(n)=sopf(n+k)?

Como la cuestión directa puede tener una respuesta complicada, abordamos antes la inversa, que dado un N debamos encontrar el K correspondiente.

Recordemos: la función SOPF(N) es la suma de los factores primos de N tomados sin repetición. Así SOPF(6)=2+3=5 y SOPF(12)=5 también, porque ambos tienen los mismos factores primos 2 y 3.

Primera cuestión: Encontrar K para un N dado

En nuestro Buscador tenemos implementada la función SOPF, por lo que esta cuestión se puede explorar previamente. Por ejemplo, para ver si el número 325 tiene el mismo valor de la función SOPF que otro posterior escribiríamos:

ES SOPF(325)=SOPF(325+N)

Con esta sencilla instrucción obtendremos varios valores de N que sumados a 325 conservan su valor de SOPF:

Solución
115
143
214
225
299
377
520
522
555

Como SOPF(325)=18, comprobaremos que, al sumarle 115, 143 o 214 permanece el mismo valor:

325+N	SOPF
440	18
468	18
539	18
550	18
624	18
702	18

Lo anterior solo nos vale para comprobar. Seguimos con nuestro estudio.

Existe una forma sencilla de evaluar la función SOPF sin necesidad de descomponer N en factores primos. Su código para Excel puede ser este:

```
Public Function sopf(n)
Dim f, a, e, g
a = n
f = 2 ‘Esta variable recorrerá los primos
e = 0 ‘Aquí se sumarán los factores primos
g = 0 ‘Recogerá los divisores primos
While f <= a
While a / f = a \ f ‘Se encuentra un factor
g = f: a = a / f ‘Tomamos nota en g y eliminamos ese factor
Wend
If g <> 0 Then e = e + g: g = 0 ‘Se suman factores en SOPF
If f = 2 Then f = 3 Else f = f + 2 ‘Se buscan nuevos factores
Wend
sopf = e
```

End Function

Con ella podemos ir descubriendo ya las coincidencias en los valores de SOPF. En la tabla podemos comprobar algunas.

N	SOPF(N)
6	5
12	5
144	5
21	10
63	10
147	10
20	7
80	7
100	7

Queda claro en ella que comparten valores de SOPF aquellos números que coinciden en sus factores primos sin contar repeticiones. Los tres primeros se basan en 2+3=5, los segundos en 3+7=10 y los últimos en 2+5=7.

Basta restar dos de ellos para tener una solución a la primera cuestión:

SOPF(6)=SOPF(6+6)=SOPF(6+138)

SOPF(21)=SOPF(21+42)=SOPF(21+126)

Esto resuelve la cuestión para un N dado y K desconocido: **los valores posibles de K para un N dado son infinitos**. Basta aumentar los exponentes de los factores primos de N y después restar.

Esto tiene una traducción a fórmula:

$$K = p_1^a \cdot p_2^b \cdot p_3^c \dots (p_1^r \cdot p_2^s \cdot p_3^t \dots - 1)$$

Es decir, serán valores válidos de K aquellos formados por la descomposición factorial de N multiplicada a su vez por un producto similar al que restamos una unidad. Los exponentes r, s y t pueden ser nulos (salvo uno, para evitar la solución K=0)

Por ejemplo, para $12=2^2*3$, K podría tener cualquiera de estas estructuras:

$2^2*3*(2^3-1)=84$, que equivaldría a

SOPF(12)=SOPF(12+84)=SOPF(96)=5

$2^2*3*(2*3-1)=60$, luego SOPF(12)=SOPF(12+60)=SOPF(72)=5

Así que la primera cuestión ya está resuelta, con la multiplicidad de soluciones de K `para cualquier N.

Segunda cuestión: Dado un valor de K, encontrar posibles valores de N

Esta cuestión es mucho más complicada, pues los valores de N no están acotados. Por ejemplo, para K=87654321, hemos encontrado un valor de N de 13761, lo que ha supuesto recorrer más de trece mil números. Efectivamente es una solución, porque

SOPF(13761)=153, ya que $13761=3^2*11*139$, y 3+11+139=153.

Y

SOPF(13761+87654321)=SOPF(87668082)=153, porque $87668082=2*3^9*17*131$, con lo que SOPF(87668082)=2+3+17+131=153.

No parece sencillo demostrar que la conjetura planteada sea verdadera. Son muchas las coincidencias entre sumas de primos, como 29+5=31+3=34 y la herencia de los valores de SOPF que vimos en la primera cuestión, por lo que se puede confiar en que para todo K exista un N que cumpla SOPF(N)=SOPF(N+K). Pero demostrarlo es otra cuestión bien distinta.

Sí podemos presentarlo como conjetura, y esto es lo que figura en la página web

https://www.primepuzzles.net/conjectures/conj_025.htm

rotulada como conjetura 25. En ella puedes consultar algunos intentos de demostración de la misma.

Si no se sabe demostrar la conjetura, sí se puede establecer una búsqueda del mínimo valor de N para cada K. Esto lo tienes publicado en http://oeis.org/A065925, y ahí remiten a la conjetura 25. Los primeros valores de N son 5, 2, 7, 4, 114, 2, 5, 8, 13, 10, 25, 4, 5, 2, 19, 16, 85, 6, 5, 5, 209, 22, 25, 3, 493, 26, 31, 4, 20, 2, 5, 32, 7, 34, 516, 12, 33, 38, 10, 10, 99, 6, 5, 44, 57, 46, 25, 6,…

Aquí podemos comprobar la conjetura con dos herramientas. Para Excel usaremos una “atrevida” función, que encontrará N para un K dado. La llamamos así porque se basa en un bucle **que puede no tener final** si la conjetura es falsa, con lo que habría que detener Excel antes de que terminara el proceso. Su código es

Function dsopf(k)
Dim d, n
d = 0
n = 2
While d = 0 ‘En este *while* radica el peligro, ya que puede no detenerse
If sopf(n) = sopf(n + k) Then d = n
n = n + 1
Wend
dsopf = d
End Function

Se supone definida ya la función SOPF.

Esta función devuelve un 0 si no encuentra ningún valor de N. En realidad no lo devolvería, porque la ejecución no podría detenerse. Se ve que hemos confiado en la conjetura.

Con esta función es fácil reproducir la lista de valores de N publicada en OEIS:

K	Mínimo N
1	5
2	2
3	7
4	4
5	114
6	2
7	5
8	8
9	13
10	10
11	25
12	4
13	5
14	2
15	19

También podemos probar con valores grandes de K, como el número de fecha de la Navidad de este año:

DSOPF(251219)=3256.

Comprobamos: SOPF(3256)=50,

SOPF(251219+3256)=SOPF(254475)=50

Parece que podemos confiar en que la conjetura sea cierta. Como Excel tiene una capacidad limitada cuando trata con números grandes, traducimos la cuestión al lenguaje PARI:

sopf(n)= my(f, s=0); f=factor(n); for(i=1, matsize(f)[1], s+=f[i, 1]); s

dsopf(k)= my(d=0,n=2);while(d==0,if(sopf(n)==sopf(n+k),d=n); n+=1);d

print(dsopf(12345678910))

Puedes copiar este código y ejecutarlo en la página

https://pari.math.u-bordeaux.fr/gp.html

Aquí tienes el resultado: 81147.

```
? sopf (n) = my (f, s = 0); f = factor (n); para (i = 1, matsize (f) [1], s + = f [i, 1]); s
dsopf (k) = my (d = 0, n = 2); while (d == 0, if (sopf (n) == sopf (n + k), d = n); n + = 1); d
imprimir (dsopf (12345678910))
81147
```

Cambiando el ejemplo 12345678910 por otro iríamos comprobando la conjetura para diversos valores.

Este número 12345678910 nos daría problemas de lentitud en Excel, pero hasta aquí sí puede con el problema:

K	Mínimo N
12345678910	81147

Por si acaso, es preferible usar PARI para evitar los problemas de la coma flotante de Excel.

Esto es lo que podemos afirmar sobre la conjetura. Hay que agradecer a Juan Carlos Amez su propuesta, pues nos ha permitido estudiar un tema interesante.

RELACIONES ENTRE PHI(N) Y TAU(N)

Función TAU

Las funciones PHI y TAU, aplicadas a un número entero positivo, tienen algo de complementarias. La segunda, TAU, cuenta los divisores de un número N. También es llamada función *divisor*, o D(x). En el caso de un número primo *p*, es claro que los divisores son *1* y *p*, luego la función TAU valdrá 2 en este caso. Igualmente, es fácil deducir que para potencias de un número primo, p^k, $TAU(p^k)=1+k$

Puedes acudir a nuestra publicación *Funciones multiplicativas*

(http://www.hojamat.es/publicaciones/multifun.pdf)

para consultar la fórmula general.

$TAU(N)=(1+a_1)(1+a_2)\ldots(1+a_k)$

a_1, a_2, $\ldots a_k$ son los exponentes de los factores primos de N.

Por ejemplo, $TAU(24)=TAU(2^3*3)=(1+3)(1+1)=8$

Efectivamente, los divisores de 24 son ocho: 1, 2, 3, 4, 6, 8, 12 y 24.

Función PHI

La función φ(n) (indicatriz o indicador de Euler) es **el cardinal del conjunto de elementos inversibles en Z_n** o bien el conjunto de números coprimos con n y menores que él contando el 1. Esta segunda versión es más clara y adecuada al estudio que vamos a iniciar: cuenta los números primos con N y menores que N, con el añadido del 1.

La función indicatriz de Euler es multiplicativa, porque si m y n son coprimos, se cumple que

φ(m). φ(n) = φ(m.n)

Su fórmula explícita es

$$\varphi(N) = N\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\dots\left(1-\frac{1}{p_k}\right)$$

(p_i son sus factores primos)

Por ejemplo, el número $18=3^2*2$ posee un valor de PHI igual a 18(1-1/2)(1-1/3)=6, Podemos comprobar que los números coprimos con 18 y menores que él son: 1, 5, 7, 11, 13 y 17. En total 6.

En los números primos ***p*** el valor de PHI(***p***)=***p-1***, como es fácil deducir.

En algunos lenguajes de programación recibe el nombre de función *totient.*

Relaciones entre TAU y PHI

Para cualquier número natural N, los números comprendidos entre 1 y N pertenecen a uno de estos tres conjuntos:

{A} Divisores de N: los cuenta la función TAU
{B} Coprimos con N incluido el 1: los cuenta la función PHI. En ambos conjuntos se encuentra el 1, lo que hace que no sean disjuntos.

{C} Resto de números: son aquellos números ***r*** que no son divisores de N ni coprimos con él: tienen un m.c.d con N que es mayor que 1 y menor que ***r***.

Por ejemplo, en el número 30, los conjuntos serían:
{A} = {1, 2, 3, 5, 6, 10, 15, 30}, pues 30=2*3*5 y TAU(30)=(1+1)(1+1)(1+1)=8
{B} = {1, 7, 11, 13, 17, 19, 23, 29}, y PHI(30)=30(1-1/2)(1-1/3)(1-1/5)=1*2*4=8

{C} = {4, 8, 9, 12, 14, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28}, que son 15 elementos.

La suma de los cardinales de los tres conjuntos es 31, porque el 1 está repetido, y 8+8+15=31.

Con este planteamiento se adivina que pueden existir varias relaciones distintas entre los tres cardinales. El primero lo recoge TAU y el segundo PHI. El tercero lo dejamos como complemento de los otros dos.

PHI=TAU

Según lo publicado en http://oeis.org/A020488, solo existen estos casos: 1, 3, 8, 10, 18, 24, 30.

Por ejemplo, en N=10,
TAU(10)=TAU(2*5)=(1+1)(1+1)=4 y PHI(10)=10(1-1/2)(1-1/5)=1*4=4.

No debemos conformarnos con lo publicado. Puedes comprobarlo con las dos versiones sencillas para el cálculo de ambas funciones que hemos preparado con el Basic de las hojas de cálculo. Para no interrumpir el estudio, las incluimos en un Anexo.

Jud McCranie da razones en esa página de por qué no hay más soluciones, y lo probó A. P. Minin 1894 . Lo comprobamos con nuestras funciones de hoja de cálculo:

N	PHI(N)	TAU(N)
1	1	1
3	2	2
8	4	4
10	4	4
18	6	6
24	8	8
30	8	8

El único número primo de la lista es 3, pues TAU(p)=2 para cualquier primo, y PHI(p)=p-1. Luego ha de ser 2=p-1 y p=3.

PHI doble de TAU

También existen pocos casos (http://oeis.org/A062516):

5, 9, 15, 28, 40, 72, 84, 90 y 120.

Con nuestras funciones tenemos:

N	PHI(N)	TAU(N)
5	4	2
9	6	3
15	8	4
28	12	6
40	16	8
72	24	12
84	24	12
90	24	12
120	32	16

TAU doble de PHI

Sólo hay dos casos:

N	PHI(N)	TAU(N)
2	1	2
6	2	4

Otros casos

Con PHI=TAU+1 parece que no hay ninguno, y con PHI+1=TAU, solo dos casos:

N	PHI(N)	TAU(N)
2	1	2
4	2	3

PHI múltiplo de TAU

Si solo tenemos en cuenta múltiplos propios, cuyo cociente es mayor que 1, nos aparecen muchas soluciones. Las primeras son:

N	PHI(N)	TAU(N)	COCIENTE
5	4	2	2
7	6	2	3
9	6	3	2
11	10	2	5
13	12	2	6
15	8	4	2
17	16	2	8
19	18	2	9
21	12	4	3
23	22	2	11
26	12	4	3
28	12	6	2
29	28	2	14
31	30	2	15
33	20	4	5
34	16	4	4

Si la relación de múltiplo es a la inversa, solo aparecen las soluciones ya vistas en las que TAU es el doble de PHI.

Por último, una curiosidad:

Pitagóricos

PHI y TAU son ambos catetos de una terna pitagórica. Solo se encuentran cinco soluciones:

20, 36, 60, 100, 300

Según el siguiente cuadro, en las ternas formadas sus elementos son múltiplos de las primitivas {3, 4, 5} o {9. 40, 41}.

	Factores(N)	TAU(N)	PHI(N)	SUMA	HIPOTENUSA
20	[2,2][5,1]	6	8	100	10
36	[2,2][3,2]	9	12	225	15
60	[2,2][3,1][5,1]	12	16	400	20
100	[2,2][5,2]	9	40	1681	41
300	[2,2][3,1][5,2]	18	80	6724	82

Esta sucesión estaba inédita, y la hemos publicado en en http://oeis.org/A308664. En esa página podrás leer un razonamiento de Giovanni Resta con el que justifica que la sucesión sea finita.

ANEXO

Listado de la función PHI (Basic de Excel)

Public Function euler(n)
Dim f, a, e
Dim es As Boolean
'Calcula la indicatriz de Euler de un número
a = n ‘Copia el valor de n
f = 2 ‘Inicia el listado de primos
e = n ‘Inicia el valor de PHI
While f <= a ‘Recorre los primos posibles
es = False ‘Variable que indica si hemos llegado a un divisor primo o no
While a / f = a \ f ‘Si es un factor, se va eliminando del valor de ***n***
a = a / f: es = True
Wend
If es Then e = e * (f - 1) / f ‘Si se ha encontrado un factor primo, se incorpora a PHI

If f = 2 Then f = 3 Else f = f + 2 ‘Busca el siguiente primo
Wend
euler = e
End Function

Listado de TAU

Es muy parecido al anterior
Public Function tau(n)
Dim f, a, e, exx

a = n ‘Copia el valor de n
f = 2 ‘Inicia el listado de primos
e = 1 ‘Inicia el valor de TAU
While f <= a ‘Recorre los primos posibles
exx = 0
While a / f = a \ f
a = a / f: exx = exx + 1 ‘Incrementa el exponente del factor primo encontrado
Wend
e = e * (1 + exx) ‘Construye TAU
If f = 2 Then f = 3 Else f = f + 2
Wend
tau = e
End Function

NÚMEROS CASI AMIGOS

Hoy repasaremos los llamados números comprometidos o casi amigos. Son dos números m y n tales que la suma de los divisores no triviales de uno coincide con el valor del otro. Así, son de ese tipo, 48 y 75, ya que la suma de divisores (función SIGMA) de 48 es 48+24+16+12+8+6+4+3+2+1=124, pero si no contamos el 1 y el propio 48 (divisores triviales) nos queda 75, que es el otro número. Recíprocamente, SIGMA(75)=124, y eliminando 75 y 1, nos queda 48.

Esta idea de divisores no triviales se recoge en la función de Chowla, que se puede definir como CHOWLA(n)=SIGMA(n)-n-1. Así que en estos números se cumple

CHOWLA(48)=75 y CHOWLA(75)=48

Es evidente que esta función tiene valor 0 si un número es primo. Esto confirma que estos números que estudiamos son todos compuestos.

Es trivial también que la función SIGMA coincide en ambos números m y n del par (en el ejemplo, 124) y que su valor es m+n+1. Este hecho se toma también como definición de números comprometidos:

$$\sigma(\mathrm{m}) = \boldsymbol{\sigma}(\mathbf{n}) = \mathbf{m} + \mathbf{n} + \mathbf{1}$$

Estos números están publicados en varios sitios.

Destacamos la de OEIS, en la que se les da el nombre de “números comprometidos”:

http://oeis.org/A005276

Betrothed (or quasi-amicable) numbers.
48, 75, 140, 195, 1050, 1575, 1648, 1925, 2024, 2295, 5775, 6128, 8892, 9504, 16587, 20735, 62744, 75495, 186615, 196664, 199760, 206504, 219975, 266000, 309135, 312620, 507759, 526575, 544784, 549219, 573560, 587460, 817479, 1000824, 1057595, 1081184,…

Están insertados por pares, por lo que son casi amigos 48 con 75, 140 con 195, y así hasta el final.

Búsqueda de números comprometidos

No es difícil encontrar estos pares de números. En la página de OEIS enlazada más arriba podéis consultar un procedimiento en PARI, pero, es tan sintético, que es preferible desarrollar una función en VBasic de

Excel, aunque se traduce fácilmente a otro lenguaje de programación.

Para cada número N, calcularemos la función SIGMA, suma de divisores y analizaremos si es mayor que N+1. Si lo es, la diferencia M=SIGMA(N)-N-1 es un candidato a pareja de N

Si SIGMA(M)=M+N+1, hemos dado con un número N del tipo buscado y M será su pareja.

La función SIGMA es muy popular. Una versión sencilla la tienes en

https://hojaynumeros.blogspot.com/2019/10/la-funcion-sigma-y-sus-traslados.html

Con ella construimos una función que nos devuelva un 0 si el número no es comprometido, o su pareja M si lo es.

Function comprometido(n)

Dim m, s, c

s = sigma(n)
If s > n + 1 Then ‘Sigma suficientemente grande

‘Si m cumple la reciprocidad, vale, Si no, devuelve un cero

m = s - n - 1: If sigma(m) = m + n + 1 Then c = m Else c = 0

End If

comprometido = c

End Function

Esta función permite reproducir fácilmente las parejas comprometidas ya publicadas. Basta organizar una búsqueda y publicar solo las que presentan un resultado distinto de cero. En Excel las primeras serían:

M	N
48	75
75	48
140	195
195	140
1050	1925
1575	1648
1648	1575
1925	1050
2024	2295
2295	2024
5775	6128
6128	5775
8892	16587
9504	20735

Era previsible que las parejas aparecieran duplicadas, por la reciprocidad interna en ellas. Se puede programar que solo se publique uno de los miembros de la pareja.

De esta función deducimos un listado sencillo en PARI:

```
for(n=1,500,a=sigma(n)-n-1;if(a>1,if(sigma(a)-a-1==n,
print1(n, ", "))))
```

Se confirma el listado:

48, 75, 140, 195, 1050, 1575, 1648, 1925, 2024, 2295, 5775, 6128, 8892, 9504, 16587, 20735,…

Cuestiones derivadas

Todos los pares conocidos, hasta 10^{10}, tienen distinta paridad.
Con esta función podemos preguntarnos cuál es el primer par de comprometidos a partir de un número, por ejemplo, un millón. Hemos organizado la búsqueda y resultan

1000824 y 1902215

También podemos interpretar esto como una secuencia cíclica de dos pasos:

CHOWLA(CHOWLA(N))=N

Existen números en los que estos ciclos son de más de dos pasos. Son los casi sociales, estudiados por

Mitchell Dickerman, como 1215571544, que da lugar a un ciclo de ocho pasos:

1215571544
1270824975
1467511664
1530808335
1579407344
1638031815
1727239544
1512587175
1215571544

Por último, estos números parecen no poseer otras propiedades, aparte de ser compuestos. Entre los primeros no hemos encontrado cuadrados, ni triangulares, o semiprimos., por ejemplo. Así que los dejamos aquí.

USO DE LA FUERZA BRUTA

El día 29/12/19 descubrí este tweet de @d_r_o_n_e:

DRONE
@d_r_o_n_e

Had a weird dream last night about
a+b+c=a*b*c/1000=d+e+f=d*e*f/1000
Had to brute force it this morning and found 444

Traducir Tweet

$$50+370+24=\frac{50\cdot 370\cdot 24}{1000}=444$$

$$32+375+37=\frac{32\cdot 375\cdot 37}{1000}=444$$

4:04 p. m. · 29 dic. 2019 · Twitter Web Client

En él podemos ver un ejemplo que puede reproducirse mediante el uso de la “fuerza bruta. Consiste en recorrer todas las variantes de un problema sin usar razonamientos ni condiciones complementarias. Es una buena estrategia comenzar con esta forma de buscar para después ir afinando resultados, explicarlos y, si es posible, justificarlos.

Uso de la herramienta Cartesius

Nuestra herramienta Cartesius también ayuda a combinar variables de todas las formas posibles, pero se hace lenta cuando nos acercamos al rango de los números de cuatro cifras. La puedes descargar desde

http://www.hojamat.es/sindecimales/combinatoria/herramientas/herrcomb.htm#cartesius

Con ella hemos intentado este planteo:

XTOTAL=3
XT=1..200
ES X1*X2*X3/1000=X1+X2+X3
CRECIENTE

Es fácil entender su significado: combinaremos tres variables, todas entre 1 y 200, de forma que se cumpla la condición pedida y después nos quedamos con las crecientes. Al cabo de más de media hora se obtuvo esta tabla, completada en sus dos últimas columnas con las dos expresiones que deben coincidir.

X1	X2	X3	X4	SUMA	PRODUCTO/1000
12	170	175		357	357
15	125	160		300	300
15	132	150		297	297
16	125	141		282	282
20	75	190		285	285
20	78	175		273	273
20	85	150		255	255
20	100	120		240	240
21	100	110		231	231
22	84	125		231	231
25	60	170		255	255
25	65	144		234	234
25	66	140		231	231
25	80	105		210	210
25	90	92		207	207
27	64	125		216	216
28	50	195		273	273
30	46	200		276	276
30	50	160		240	240
30	65	100		195	195

Observamos, y lo confirmaremos un poco más abajo, que aparecen repetidas algunas soluciones menores que 444, como son 231 o 240. Con el planteo propuesto se encontraron 32 soluciones, de las que solo hemos reproducido las primeras. Aparecen en orden inverso al natural porque cuando unos factores tienen igual suma, su producto crece cuando sus diferencias son menores. Con este intento descubrimos ya que la técnica de la “fuerza bruta” es muy lenta en producir resultados.

Analizando la búsqueda descubrimos que Cartesius ha tenido que analizar 200*200*200 =8*10^6 números, y en cada uno calcular si una igualdad se verifica o no. Eso es mucho para un portátil normal. Ahí es donde falla la fuerza bruta, en la multiplicación de casos que produce la Combinatoria.

Algoritmos

La “fuerza bruta” se caracteriza casi siempre por el uso de bucles del tipo FOR_NEXT, WHILE o REPEAT, casi siempre anidados en tres o cuatro niveles.

Lo normal, en ejemplos similares al que nos ocupa, es disponer de tres bucles anidados, con la propiedad deseada en el interior de los tres. Comenzaremos exigiendo solo una condición de las propuestas por @d_r_o_n_e

En este caso podíamos comenzar por este código:

Sub fuerzabuta()
Dim i, j, k, a, b, fila

fila = 10 ‘La fila determina la construcción de una tabla en Excel
For i = 1 To 1000 ‘Bucle triple
For j = 1 To i
For k = 1 To j
a = i + j + k ‘Cálculos previos
b = i * j * k / 1000
If a = b Then ‘Condición pedida
fila = fila + 1: ‘Construcción de la tabla
ActiveWorkbook.ActiveSheet.Cells(fila, 2).Value = b
ActiveWorkbook.ActiveSheet.Cells(fila, 3).Value = i
ActiveWorkbook.ActiveSheet.Cells(fila, 4).Value = j
ActiveWorkbook.ActiveSheet.Cells(fila, 5).Value = k
End If

Next k
Next j
Next i
End Sub

Hemos ejecutado esta macro de Excel y nos han resultado muchas soluciones. Lo que nos interesa es que salgan repetidas. Por la forma de plantear el problema, aparecerán desordenadas. Las primeras han sido:

Número	i	j	k
165	60	55	50
168	70	50	48
171	76	50	45
186	80	75	31
189	84	75	30
180	90	50	40
207	92	90	25
189	100	54	35
192	100	60	32
195	100	65	30
210	105	80	25
231	110	100	21
204	120	50	34

Coinciden con las obtenidas en Cartesius. Observamos su falta de orden y la existencia de un repetido, el 189. Según la tabla:

189=84+75+30=84*75*30/1000

189=100+54+35=100*54*35/1000

Es una solución también más pequeña que la propuesta de 444.

Para verlas todas ordenaremos la columna y así se verán mejor los repetidos. Con este método hemos descubierto los siguientes: 189, 207, 231, 240, 255, 273, 297, 420, 444, 480, 504, 741, 759, 768, 810, 891,… De ellos presentan soluciones triples 231, 504 y 891.

Más fuerza bruta (o menos)

Podíamos intentar descubrir tan solo los números en los que se da más de una solución. El problema es que para esto se necesitaría un bucle más, con el consiguiente aumento de tiempo de proceso. Es el coste de utilización de bucles múltiples sin apenas condicionamientos. Hay una forma de evitar un nuevo bucle, y es considerar que en el algoritmo anterior hemos hecho variar el valor de **k** cuando en realidad está condicionado por la igualdad que se pide ***a=i+j+k.*** Considerándolo así, lo único que ha de cumplir k es que su valor sea ***a-i-j*** y, por cuestión de unicidad, que no sea mayor que **j**. Esto es lo que hemos implementado en PARI:

```
for(n=1,500,m=0;b=0;for(i=1,n,for(j=1,i,c=i+j;if(c<=n&&n-c<=j,b=i*j*(n-c)/1000;if(b==n,m+=1))));if(m>1,print1(n,", ")))
```

Mantenemos los bucles con las variables i y j. Eliminamos el bucle de k sustituyéndolo por la expresión ***b=i*j*(n-c)/1000*** y concretamos las condiciones que ha de cumplir. Con la variable **m** exigimos que haya repetición de casos. Hemos añadido un nuevo bucle, pero con los cambios apenas se resiente la velocidad del proceso. De todas formas, para rangos de números de 1000 más o menos, puede tardar muchos minutos o incluso más de una hora. Cosas de la fuerza bruta.

Los primeros resultados son:

189, 207, 231, 240, 255, 273, 297, 420, 444, 480, 504, 741, 759, 768, 810, 891, 1221, 1320, 2418,…

Con esto damos por terminada la búsqueda, porque la fuerza bruta cansa y no se aprende mucho con ella.

Rebajamos pretensiones

Podíamos exigir productos similares, pero con solo dos variables, es decir, N=i+j=i*j/100. Esto simplifica el problema, y solo lo incluimos como repaso de las técnicas empleadas anteriormente. Las explicaciones para el caso anterior valen también para este.

Con Cartesius

Plantearíamos, por ejemplo:

xtotal=2
xt=1..700
es x1+x2=x1*x2/100
creciente

Obtendríamos:

X1	X2	X3	X4	X5
120	600		720	720
125	500		625	625
140	350		490	490
150	300		450	450
180	225		405	405
200	200		400	400

Las primeras columnas corresponden a los valores de i, j y las siguientes el valor repetido de N.

Así, 120+600=120*600/100=720

125+500=125*500/100=625

En principio, no parece que existan soluciones dobles.

Con una función de Excel:

Function esmultiple2(n)
Dim i, j, k, a, b, m

m = 0
For i = 1 To n

For j = 1 To i

a = i + j

b = i * j / 100

If a = b And a = n Then m = m + 1

Next j

Next i

esmultiple2 = m

End Function

Esta función cuenta las veces en las que se da la igualdad ***i+j=i*j/100***. Organizando una búsqueda nos resulta:

400
405
450
490
625
720
841
1210
1458

Tampoco se aprecian repetidos

Con PARI

Traducimos la función anterior a PARI y la integramos en un bucle de búsqueda:

for(n=1,10000,m=0;b=0;for(i=1,n/2,b=i*(n-i)/100;if(b==n,m+=1));if(m>0,print1(n,", ")))

Volvemos a obtener los mismos resultados:

400, 405, 450, 490, 625, 720, 841, 1210, 1458, 2205, 2704, 5202,…

Tampoco aquí se detectan repetidos. Lo dejamos como complemento.

SUMA Y PRODUCTO DE CUBO Y OTRO TIPO

Muchas cuestiones en este documento surgen de los cálculos diarios que publico en Twitter (@Connumeros).

El día 21/5/19 obtuve esta propiedad:

El número de fecha de hoy, 21519 se descompone en un producto de un cubo por un capicúa y también en una suma del mismo tipo:

21519=3^3×797

21519=12^3+19791

No sabía en ese momento si existirían muchos números que compartieran las dos expresiones N=p^3*q y N=r^3+s, y parece que sí, que son abundantes.

Para acotar la búsqueda, exigiremos que los cuatro números p, q, r y s sean enteros positivos. La exclusión del 0 evita casos triviales.

Al iniciar el estudio he pensado que el número que acompaña al cubo puede ser cuadrado o triangular, por ejemplo, en lugar de capicúa.

Suma y producto de cubo y capicúa

La primera condición, N=p^3*q, permite desechar aquellos números N que no sean múltiplos de un cubo. Esto se logra fácilmente con la descomposición factorial y el estudio de los exponentes de los factores primos. El inconveniente es que se alargaría mucho la explicación del procedimiento para crear nuestra función FACTORES y la rutina SACAPRIMOS. Por eso, y no es nuevo en nuestros documentos, emprenderemos la búsqueda con medios más sencillos. El peligro estribaría en la lentitud, pero no es inconveniente en este caso. Con Excel se consiguen listas con una rapidez aceptable.

Comenzamos, como es usual en estas búsquedas, con la creación de una función, a la que llamaré CUBOYOTRO, que nos indique si un número N cumple los dos requisitos N=p^3*q y N=r^3+s. Su estructura nos va a permitir adaptarla a todos los casos que estudiemos, pues bastará sustituir la función ESCAPICUA (para el caso inicial) por ESCUAD, ESTRIANGULAR u otra. En cada tipo explicaremos estas funciones auxiliares. Comenzamos con los capicúas. La función recomendada es la siguiente:

Public Function cuboyotro$(n, k) ‘Añadimos un parámetro k por si deseamos cambiar cubo por otra potencia

Dim x, a, y, b, c

Dim s$

s$ = "" ‘Usamos un string para presentar bien los cuatro números p, q, r y s

a = n ^ (1 / k) ‘En este primer caso k valdrá 3. La variable a es el tope de búsqueda

For x = 1 To a

c = n - x ^ k ‘Se resta del número la potencia (en el primer ejemplo, un cubo)

If escapicua(c) And c > 10 Then ‘Más adelante se cambiará ESCAPICUA

For y = 2 To a ‘En esta parte ya se ha cumplido la segunda condición N=r^3+s

If n Mod y ^ k = 0 Then ‘Para la primera condición p^3 ha de ser un divisor de n

b = n / y ^ k

If escapicua(b) and b>10 Then ‘Si el cociente es capicúa, se publica la solución

s$ = " C1 " + Str$(x) + " O1 " + Str$(c) + " C2 " + Str$(y) + " O2 " + Str$(b)

‘El string nos presenta los cubos C1 y C2 y sus compañeros O1 y O2.Puede haber más soluciones.

End If

End If

Next y

End If
Next x
If s$ = "" Then s$ = "NO" ‘Asignamos un “NO” al caso sin solución
cuboyotro = s$
End Function

Hay que advertir algún detalle sobre esta función.

La decisión de evaluar en primer lugar la segunda condición y después la primera no ha sido deliberada, y de hecho, poco eficiente, pues si se cambia el orden se incrementa la velocidad de respuesta de la función. Como resulta rápida así, no se ha corregido y lo dejamos como ejercicio.

Este esquema es la base para otras búsquedas. Ya se ha destacado que con un cambio de ESCAPICUA por otra función se podrían abordar otros casos. Igualmente, aunque en lo que sigue haremos k=3 para buscar cubos, se deja abierta la posibilidad de aumentar el exponente.

La función ESCAPICUA se inserta en el Anexo del final de este capítulo. La costumbre es considerar capicúas los números de una cifra, pero aquí no nos interesa esta posibilidad, pues aparecen casos sin interés. Exigiremos que sean mayores que 10, como puedes comprobar en el listado de la función.

Los primeros números con esta propiedad son

528	C1 4 O1 464 C2 2 O2 66
704	C1 2 O1 696 C2 4 O2 11
888	C1 7 O1 545 C2 2 O2 111
1128	C1 8 O1 616 C2 2 O2 141
1188	C1 8 O1 676 C2 3 O2 44
1208	C1 8 O1 696 C2 2 O2 151
1375	C1 11 O1 44 C2 5 O2 11
1408	C1 11 O1 77 C2 4 O2 22
1616	C1 10 O1 616 C2 2 O2 202
1696	C1 10 O1 696 C2 2 O2 212
1856	C1 11 O1 525 C2 2 O2 232
2176	C1 4 O1 2112 C2 2 O2 272
2424	C1 12 O1 696 C2 2 O2 303
2727	C1 12 O1 999 C2 3 O2 101
2904	C1 13 O1 707 C2 2 O2 363
2984	C1 13 O1 787 C2 2 O2 373
3064	C1 8 O1 2552 C2 2 O2 383
3552	C1 14 O1 808 C2 2 O2 444
3632	C1 14 O1 888 C2 2 O2 454
3773	C1 11 O1 2442 C2 7 O2 11

Junto a cada uno se presentan los cubos C1 y C2 y el otro componente, en este caso capicúa, en O1 y O2.

Por ejemplo, 2176=4^3+2112=2^3*272, dos cubos y dos capicúas.

En PARI

Al tener que cumplir varias condiciones, el listado para PARI resulta algo extenso, pero es bastante rápido en su ejecución.

maxexpo(n) = s=1; f=factor(n); for(i=1, matsize(f)[1], t=f[i,2]; if(t>>s, s=t)); s

palind(n)=n==eval(concat(Vecrev(Str(n))))

condi1(n)= c=0; if(maxexpo(n)>=3, a=n^(1/3); for(x=2, a, if(n%x^3==0,b=n/x^3;if(palind(b)&&b>=10,c=x))));c

condi2(n)= c=0; a=n^(1/3); for(x=2, a, b=n-x^3;if(palind(b)&&b>=10,c=x));c

for(y=2,20000, if(condi1(y)&&condi2(y),print1(y,", ")))

Con él podemos reproducir y ampliar la lista de arriba:

528, 704, 888, 1128, 1208, 1375, 1408, 1616, 1696, 1856, 2176, 2424, 2727, 2904, 2984, 3064, 3552, 3632, 3773, 3952, 4280, 4347, 4440, 4520, 4752, 5488, 5568, 5736, 5994, 6296, 6336, 6464, 6784, 7352, 7752, 8181, 8384, 8888, 10071, 10944, 11000, 11264, 11319, 12224, 12798, 13635, 13875, 14168, 14641, 15928, 16128, 16362, 16375, 17172, 18048, 18656, 19008, 19536, 19629, 19899,…

Hay que recordar que todos ellos son múltiplos de un cubo con base no trivial y, por tanto, todos son compuestos. Entre ellos aparecen casos particulares interesantes. Por ejemplo:

Números del tipo p^3*11 o p^3*101. En estos dos casos y otros similares, el capicúa correspondiente al producto es un número primo, como ocurre en

704=4^3*11=2^3+696.

Caso del 14641: Como equivale a 11^4, su desarrollo sería 11^3*11. Hay que esperar que pertenezcan a este listado potencias de primos, aunque sin buscarlos no se puede asegurar. Por ejemplo, 101^4 cumple la primera condición (producto), pero no es suma de cubo y capicúa. El siguiente es 40353607, que es potencia de primo (40353607=7^9) y se descompone en producto de cubo y capicúa (40353607=49^3*343) y en suma de cubo y capicúa (40353607=334^3+3093903). Hasta una cota de 8*10^7 ya no hay más casos.

El número 14641 es capicúa. Podríamos preguntarnos si existen más capicúas en la sucesión. En la primera tabla hemos visto algún capicúa. Los primeros son: 888, 3773, 6336, 8888, 14641, 80008, 88088,…

Por ejemplo, 3773 es capicúa, y equivale a 11^3+2442 y a 7^3*11.

Igualmente, el capicúa 6336 es igual a 11^3+5005 y a 4^3*99.

Finalmente, destacamos el número 74088, que es el cubo de 42, y también coincide con la suma de otro cubo y un capicúa, 35^3+31213, y también con un producto similar, 6^3*343. Esto es posible por ser 343 capicúa y cubo de 7.

Se podría buscar más casos particulares, pero es preferible pasar a otras estructuras, que dejaremos para el siguiente apartado.

ANEXO

Código de la función ESCAPICUA

Public Function escapicua(n) As Boolean

```
Dim l, i, k
Dim c As Boolean
Dim auxi$,nn$

nn$ =Str$(n)
auxi= Right(nn$, Len(nn$) - 1)
l = Len(auxi)
c=True
If l >1 Then
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
```

Seguimos con el tema

Hemos estudiado hasta ahora los números que son suma y también producto de un cubo y un capicúa. En esta buscaremos casos similares con cuadrados y triangulares.

Caso cubo y cuadrado

Tal como anunciamos en el apartado anterior, si sustituimos ESCAPICUA en la función CUBOYOTRO por ESCUAD, que determina si un número es cuadrado perfecto, podríamos repetir el estudio para cuando los factores y sumandos fueran uno cubo y otro cuadrado. El listado de esta otra función puede ser el siguiente:

```
Public Function escuad(n) As Boolean
If n < 0 Then
escuad = False
Else
If n = Int(Sqr(n)) ^ 2 Then escuad = True Else escuad
= False
End If
End function
```

Efectuando la sustitución, resultan los números de la tabla, como los menores que cumplen las condiciones exigidas:

72	C1	2	O1	64	C2	2	O2	9
108	C1	3	O1	81	C2	3	O2	4
128	C1	4	O1	64	C2	2	O2	16
392	C1	7	O1	49	C2	2	O2	49
512	C1	7	O1	169	C2	2	O2	64
576	C1	8	O1	64	C2	4	O2	9
968	C1	7	O1	625	C2	2	O2	121
1323	C1	3	O1	1296	C2	3	O2	49
1372	C1	6	O1	1156	C2	7	O2	4
1568	C1	7	O1	1225	C2	2	O2	196
1944	C1	2	O1	1936	C2	6	O2	9
2000	C1	4	O1	1936	C2	5	O2	16
2304	C1	12	O1	576	C2	4	O2	36
2312	C1	2	O1	2304	C2	2	O2	289
2700	C1	11	O1	1369	C2	3	O2	100
2888	C1	14	O1	144	C2	2	O2	361
3200	C1	4	O1	3136	C2	2	O2	400
3267	C1	11	O1	1936	C2	3	O2	121

Ejemplo: 1323=3^3+36^2=3^3*7^2

Con PARI hay que cambiar un poco el algoritmo, por las peculiaridades de la función *issquare*:

```
condi1(n)= my(c=0); a=truncate(n^(1/3)); for(x=2, a, for(b=2,sqrt(n),if(n==x^3*b^2,c=1)));c
condi2(n)= my(c=0); a=truncate(n^(1/3)); for(x=1, a, b=n-x^3;if(issquare(b)&&b>0,c=x));c
for(y=1,20000, if(condi1(y)&&condi2(y),print1(y,", ")))
```

Así podemos ampliar el listado anterior:

72, 108, 128, 392, 512, 576, 968, 1323, 1372, 1568, 1944, 2000, 2304, 2312, 2700, 2888, 3200, 3267, 3456, 3528, 4000, 4608, 5400, 6272, 6400, 6561, 6912, 8192, 8748, 9000, 9800, 10125, 10952, 12168, 12348, 12544, 14283, 14400, 16200, 16928, 17496, 18000, 18252, 18496, 19773,…

La simultaneidad de un cubo y de un cuadrado en un producto hace sospechar que algunos términos sean potencias perfectas en esta sucesión. En efecto, los primeros casos son:

128=2^7, 512=2^9, 6561=3^8, 8192=2^13, …

En ellos el exponente se ha formado combinando el 3 del cubo con el 2 del cuadrado.

Caso cubo y triangular

En la función CUBOYOTRO podemos sustituir la función ESCUAD por la ESTRIANGULAR. Un número es triangular cuando al multiplicarlo por 8 y sumar 1 se convierte en cuadrado. Lo puedes ver con un sencillo desarrollo:

$8*T(n)+1 = 8*n*(n+1)/2+1 = 4n^2+4n+1 =(2n+1)^2$

Con esta propiedad se construye un criterio para saber si un número es triangular:

Function estriangular(n) As Boolean
Dim a
If escuad(8 * n + 1) Then estriangular = True Else estriangular = False
End Function

Sustituimos en CUBOYOTRO la función ESCAPICUA (o ESCUAD) por esta otra y obtendremos los números que son producto de cubo y triangular y también una suma del mismo tipo. Los primeros son:

8	C1	2	CAP1	0	C2	2	CAP2	1
27	C1	3	CAP1	0	C2	3	CAP2	1
48	C1	3	CAP1	21	C2	2	CAP2	6
405	C1	3	CAP1	378	C2	3	CAP2	15
567	C1	8	CAP1	55	C2	3	CAP2	21
648	C1	8	CAP1	136	C2	6	CAP2	3
750	C1	9	CAP1	21	C2	5	CAP2	6
960	C1	9	CAP1	231	C2	4	CAP2	15
1029	C1	9	CAP1	300	C2	7	CAP2	3
1215	C1	8	CAP1	703	C2	3	CAP2	45
1344	C1	6	CAP1	1128	C2	4	CAP2	21
1680	C1	3	CAP1	1653	C2	2	CAP2	210
1848	C1	12	CAP1	120	C2	2	CAP2	231
2024	C1	2	CAP1	2016	C2	2	CAP2	253

Como en anteriores ocasiones, C1 y C2 son los dos cubos y CAP1, CAP2, en este caso, los triangulares (se ha deslizado la abreviatura de capicúa).

Por ejemplo, 1029=9^3+300=9^3+24*25/2, suma de cubo y triangular, y además, 1029=7^3*3=7^3*2*3/2. Producto de cubo y triangular.

En estos ejemplos está incluido el 0 como triangular. En el siguiente listado, obtenido con PARI, no figuran:

48, 405, 567, 648, 750, 960, 1029, 1215, 1344, 1680, 1848, 2024, 2106, 2160, 2835, 2880, 3240, 3248, 3430, 3480, 3672, 4760, 5145, 5328, 5670, 7203, 8100, 8232, 10125, 12160, 12320, 12555, 13392, 15000, 15147, 15309, 15435, 15624, 16128, 16848, 17982, 18865, 19656,…

Con vistas a estudiar este lenguaje, se inserta el código usado:

condi1(n)= my(c=0); a=truncate(n^(1/3)); for(x=2, a, for(b=2,sqrt(2*n),if(n==x^3*b*(b+1)/2,c=1)));c

condi2(n)= my(c=0); a=truncate(n^(1/3)); for(x=1, a, b=n-x^3;if(issquare(8*b+1)&&b>0,c=x));c

for(y=1,20000, if(condi1(y)&&condi2(y),print1(y,", ")))

Cubos con primos

Para esta modalidad necesitamos la función ESPRIMO. La puedes consultar, por ejemplo, en la dirección

https://hojaynumeros.blogspot.com/2016/05/palprimos-primos-palindromicos.html

Al igual que se procedió en casos anteriores, sustituimos ESCAPICUA por ESPRIMO en la función CUBOYOTRO, con el resultado:

24	C1 1 PR1 23 C2 2 PR2 3
40	C1 3 PR1 13 C2 2 PR2 5
54	C1 1 PR1 53 C2 3 PR2 2
56	C1 3 PR1 29 C2 2 PR2 7
81	C1 4 PR1 17 C2 3 PR2 3
88	C1 3 PR1 61 C2 2 PR2 11
104	C1 1 PR1 103 C2 2 PR2 13
128	C1 5 PR1 3 C2 4 PR2 2
135	C1 4 PR1 71 C2 3 PR2 5
136	C1 5 PR1 11 C2 2 PR2 17
152	C1 1 PR1 151 C2 2 PR2 19
184	C1 5 PR1 59 C2 2 PR2 23
189	C1 2 PR1 181 C2 3 PR2 7
192	C1 5 PR1 67 C2 4 PR2 3
232	C1 5 PR1 107 C2 2 PR2 29
250	C1 3 PR1 223 C2 5 PR2 2
296	C1 3 PR1 269 C2 2 PR2 37
297	C1 4 PR1 233 C2 3 PR2 11

Si observamos las dos últimas filas, descubriremos muchos números primos como base del segundo cubo. En este caso, el número tendrá una descomposición en factores primos del tipo N=p^3*q, con lo que poseerá ocho divisores si p es distinto de q, porque TAU(N)=(3+1)(1+1)

Por ejemplo, 189=2^3+181=3^3*7, y sus ocho divisores son 189, 63, 27, 21, 9, 7, 3 y 1.

Si p=q, N=p^4, como es el caso de 81, y TAU(81)=1+4=5, siendo sus divisores 81, 27, 9, 3 y 1.

Terminamos aquí los casos. Podríamos ahora repetir el trabajo con cuartas o quintas potencias, pero se intuye que no tendrían demasiado interés. Como propuesta, se incluyen los primeros de algunos casos:

Potencias cuartas con primos

Número	Descomposición
32	C1 1 PR1 31 C2 2 PR2 2
48	C1 1 PR1 47 C2 2 PR2 3
80	C1 1 PR1 79 C2 2 PR2 5
112	C1 3 PR1 31 C2 2 PR2 7
208	C1 3 PR1 127 C2 2 PR2 13
243	C1 2 PR1 227 C2 3 PR2 3

Por ejemplo, 112=3^4+31=2^4*7

Potencias cuartas con cuadrados

Número Descomposición

400 C1 4 PR1 144 C2 2 PR2 25
2025 C1 6 PR1 729 C2 3 PR2 25
3600 C1 6 PR1 2304 C2 2 PR2 225
6400 C1 8 PR1 2304 C2 4 PR2 25
15625 C1 10 PR1 5625 C2 5 PR2 25

Así, 3600=6^4+48^2=2^4*15^2

Es fácil razonar que todos los números de este tipo son cuadrados.

Potencias cuartas con triangulares

16 C1 1 PR1 15 C2 2 PR2 1
96 C1 3 PR1 15 C2 2 PR2 6
2401 C1 4 PR1 2145 C2 7 PR2 1
3040 C1 5 PR1 2415 C2 2 PR2 190

No tiene interés seguir con más ejemplos. Aquí terminamos.

ÁREAS DE TRIÁNGULOS PITAGÓRICOS

El 18 de septiembre de 2009 publiqué una de mis primeras entradas de mi blog, con el siguiente breve texto:

Ternas pitagóricas que comparten área

La lectura de la biografía de Lewis Carroll me ha sugerido el proponer la siguiente búsqueda, inspirada en un problema que le impidió dormir una noche:

¿Qué números enteros equivalen al área de un triángulo rectángulo de lados también enteros, de tres formas distintas?

La primera solución es 840, porque las tres ternas

15, 112 y 113

24, 70 y 74

40, 42 y 58

pertenecen a lados de triángulos rectángulos de área 840.

¿Cuáles son las siguientes soluciones?

A partir de ella, mi amigo Claudio Meller (https://twitter.com/MellerClaudio) publicó en OEIS el resto de soluciones, como puedes comprobar en http://oeis.org/A177021

A partir de cálculos que no vienen al caso, me ha apetecido volver a este tema, estudiando otros casos parecidos y los procedimientos para llegar a ellos.

Procedimiento general de búsqueda

Las condiciones del problema se traducen, dado un número N, en encontrar los dos catetos de una terna pitagórica adecuada, sea $a^2+b^2=c^2$, tales que a*b=2*N. Bastará entonces buscar pares de divisores de 2N que sean catetos en una terna pitagórica. La hipotenusa c no tiene por qué intervenir.

Si organizamos la búsqueda con estas propiedades, será útil contar los pares válidos, para ver a cuántas áreas equivale N. También, según decidimos últimamente, podemos crear una función que devuelva una cadena de texto con los valores de los catetos. Proponemos esta, para Excel o Calc:

Function areapitag$(n)
Dim p, q, m
Dim s$
s$ = "": m = 0 ‘Se pone a cero el contador y el resultado
For p = 2 To Sqr(2 * n) ‘Un cateto no puede sobrepasar la raíz cuadrada de 2N
q = 2 * n / p ‘El otro divisor
If q = Int(q) Then ‘Tiene que ser entero

‘Si es terna pitagórica, se memoriza y se incrementa el contador

If escuad(p ^ 2 + q ^ 2) Then m = m + 1: s$ = s$ + " # " + Str$(p) + ", " + Str$(q)

End If

Next p

If s$ = "" Then s$ = "NO" Else s$ = Str$(m) + s$

areapitag = s

End Function

Así, si tomamos, por ejemplo el 840 del texto de arriba, nos devolverá:

AREAPITAG(840)=” 3 # 15, 112 # 24, 70 # 40, 42”

Significará que hay tres soluciones (primer 3 de la cadena) y que los catetos de cada una son (15,112), (24,70) y (40, 42), tal como afirmamos hace más de diez años.

Con esta función, si leemos el primer dígito del resultado S$, sabremos cuántas soluciones presenta cada número dado. En Excel podemos usar LEFT$(S;2), sabiendo el número está precedido por un espacio en blanco, o bien MID$(S,2,1). En cualquier bucle de búsqueda que organicemos, usaremos una de estas dos condiciones para identificar qué números no presentan solución o bien una o más de una.

Antes de emprender búsquedas, hay que advertir que si un número **a** posee un número de soluciones, también las tendrá $a*m^2$, por la generación de las distintas ternas como múltiplos de una terna primitiva.

Números que son área de al menos una terna

La primera cuestión puede consistir en descubrir qué números coinciden con al menos un área de triángulo pitagórico. Con la función anterior *areapitag* basta exigir que el resultado sea distinto de "NO". Los primeros números de este tipo son:

6	1 # 3, 4	
24	1 # 6, 8	
30	1 # 5, 12	
54	1 # 9, 12	
60	1 # 8, 15	
84	1 # 7, 24	
96	1 # 12, 16	
120	1 # 10, 24	
150	1 # 15, 20	
180	1 # 9, 40	
210	2 # 12, 35	# 20, 21
216	1 # 18, 24	
240	1 # 16, 30	
270	1 # 15, 36	
294	1 # 21, 28	
330	1 # 11, 60	
336	1 # 14, 48	
384	1 # 24, 32	

Todos equivalen a un área, salvo 210 que admite dos. Están ya publicados, como algunos de los que estudiaremos hoy.

Areas of Pythagorean triangles: numbers which can be the area of a right triangle with integer sides.

6, 24, 30, 54, 60, 84, 96, 120, 150, 180, 210, 216, 240, 270, 294, 330, 336, 384, 480, 486, 504, 540, 546, 600, 630, 720, 726, 750, 756, 840, 864, 924, 960, 990, 1014, 1080, 1176, 1224, 1320, 1344, 1350, 1386, 1470, 1500, 1536, 1560, 1620, 1710, 1716, 1734, 1890

http://oeis.org/A009112

Todos los términos son múltiplos de 6.

Se generan en PARI con un código algo oscuro. Es preferible este otro que proponemos, copia de la función *areapitag*:

for(i=1,2000,m=0;for(p=2,sqrt(2*i),q=2*i/p;if(q==truncate(q)&&issquare(p^2+q^2),m+=1));if(m>0,print1(i,", ")))

Puedes comprobar que se llega al mismo listado.

De ellos, algunos solo admiten una representación, como se ha visto en la tabla de más arriba. Si en el código PARI sustituimos ***m>0*** por ***m==1***, los obtendremos:

6, 24, 30, 54, 60, 84, 96, 120, 150, 180, 216, 240, 270, 294, 330, 336, 384, 480, 486, 504, 540, 546, 600, 630, 720, 726, 750, 756, 864, 924, 960, 990, 1014, 1080, 1176, 1224, 1344, 1350, 1386, 1470, 1500, 1536, 1560, 1620, 1710, 1716, 1734, 1920, 1944,…

Observamos que ya no está 210, que equivale a dos áreas. Esta sucesión no figura en OEIS.

Si en el código cambiamos ***m==1*** por ***m==2***, obtendremos los números que equivalen a dos áreas.

210, 1320, 1890, 2730, 4914, 5250, 5280, 7980, 10290, 11880, 17010, 18480, 19656, 21120, 24570, 25410, 29400, 30600, 32130, 33000, 34650, 35490, 41580, 44226,…

Por último, si igualamos a 3, resultará la sucesión con la que comenzamos este estudio

A177021 Numbers which are the area of exactly three Pythagorean triangles.

840, 3360, 7560, 10920, 13440, 21000, 30240, 31920, 41160, 43680, 53760, 68040, 84000, 98280, 101640, 120960, 127680, 141960, 164640, 166320, 174720, 189000, 215040, 242760, 272160, 273000, 286440, 287280, 303240, 336000, 370440, 393120, 406560, 444360

AUTHOR Claudio Meller, on a suggestion by Antonio Roldán, Dec 08 2010

Para finalizar, si deseas practicar un poco, intenta encontrar estos números con nuestra función *areapitag* (con Excel o PARI). En esta sucesión a(n) es el menor número que equivale a las áreas de n triángulos pitagóricos:

A055193 Smallest number that is the area of n distinct Pythagorean triangles.

6, 210, 840, 341880, 71831760, 64648584000, 2216650756320, 22861058133513600

PRODUCTOS CÍCLICOS CON NÚMEROS PRIMOS:

Hace unos meses estudiamos el tipo de expresión N=a*b + b*c + c*a, a la que llamamos "productos cíclicos". Puedes leerla en

https://hojaynumeros.blogspot.com/2019/03/productos-ciclicos.html

En esa entrada se estudió la unicidad de esta representación para algunos números y aquellos otros que no la admiten para ningún valor. Llegamos a algunas sucesiones finitas ya publicadas. En esta de hoy nos limitaremos al uso de tres números primos distintos.

De entrada se puede razonar que todos los números que consideraremos serán impares, ya que si en a*b + b*c + c*a, a, b y c son primos, puede ocurrir que uno de ellos sea 2, con lo que se sumarán dos productos pares y uno impar, y si ninguno es igual a 2, los tres sumandos serán impares, y también la suma lo será.

En las búsquedas previas que hemos emprendido se ha visto que existen muchos casos distintos en unicidad y número de soluciones. Por ello diseñaremos una función similar a la usada en la entrada enlazada, *Function prodciclo$(n)*, pero que solo admita factores primos distintos y que devuelva los ciclos encontrados y el número de ellos. De esta forma podremos establecer las búsquedas que deseemos. La denominaremos *prodcicloprim$.* Su esquema es parecido a la anterior, ya que recorremos todos los primos posibles, y de cada par calculamos el tercero a partir de N. Si resulta ser entero, primo y menor que los otros dos, ya hemos encontrado los tres primos buscados. En ese caso se recoge el resultado y se van contando las soluciones, Su algoritmo en VBasic de Excel puede ser:

```
Public Function prodcicloprim$(n)
Dim s$
Dim i, j, k, m

s$ = "": m = 0   ‘s$ recogerá resultados y m los contará
For i = 2 To (n - 2) / 2 ‘Primer primo
If esprimo(i) Then
j = 2
While j < i And j < n / i ‘Segundo primo
If esprimo(j) Then
```

k = (n - i * j) / (i + j) ‘Tercer posible primo

‘Si **k** cumple los requisitos, lo incorporamos a la solución e incrementamos el contador **m**

If k = Int(k) And k < j And esprimo(k) Then m = m + 1: s$ = s$ + " -- " + Str$(i) + Str$(j) + Str$(k) + " "

End If

j = j + 1

Wend

End If

Next i

If s$ <> "" Then prodcicloprim = Str$(m) + " " + s$ Else prodcicloprim = "NO"

End Function

Con esta función podremos buscar los números que permiten esta descomposición. Bastará que la misma no devuelva “NO”. También podremos contar soluciones, ya que la respuesta comienza con ese número. Por ejemplo:

prodcicloprim(191)= 4 -- 13 7 5   -- 13 11 2   -- 17 7 3   -- 37 3 2

Esta respuesta nos indica que existen 4 soluciones, que son

191=13*7+7*5+5*13

191=13*11+11*2+2*13

191=17*7+7*3+3*17

191=37*3+3*2+2*37

Con un poco de experiencia en búsquedas se le puede sacar mucho partido a esta respuesta. Según las necesidades, podemos alterar el código para que solo nos devuelva el número de soluciones, o solo estas. Ya dependerá de nuestros intereses. Por ejemplo, el día 10/01/20 publiqué en Twitter que 311 es el primer número que admite ocho descomposiciones de este tipo:

311 es el menor número que es igual a ocho expresiones de la forma pq+qr+rp, con p, q y r primos distintos:

311=13×11+11×7+7×13
311=17×13+13×3+3×17
311=19×13+13×2+2×19
311=23×7+7×5+5×23
311=29×7+7×3+3×29
311=37×5+5×3+3×37
311=43×5+5×2+2×43
311=61×3+3×2+2×61

Con esta función emprenderemos las búsquedas que deseemos:

Números que admiten al menos una representación de este tipo

Exigimos que **prodcicloprim** sea distinta de “NO”:

Nos resulta una sucesión que ya está publicada:

31	1 -- 5 3 2		
41	1 -- 7 3 2		
59	1 -- 7 5 2		
61	1 -- 11 3 2		
71	2 -- 7 5 3	-- 13 3 2	
87	1 -- 11 5 2		
91	1 -- 17 3 2		
101	2 -- 13 5 2	-- 19 3 2	
103	1 -- 11 5 3		
113	1 -- 11 7 2		
119	1 -- 13 5 3		
121	1 -- 23 3 2		
129	1 -- 17 5 2		
131	2 -- 11 7 3	-- 13 7 2	
143	1 -- 19 5 2		
151	3 -- 13 7 3	-- 17 5 3	-- 29 3 2
161	1 -- 31 3 2		
167	3 -- 11 7 5	-- 17 7 2	-- 19 5 3

En la tabla figuran los primeros números que admiten la expresión y junto a ellos el número de soluciones y los primos correspondientes. Vemos números con una, dos o tres representaciones. En cuanto se avanza algo más aparecen más casos múltiples, como el citado 311.

Puedes consultar http://oeis.org/A238397

Números que no admiten esta descomposición

Si buscamos los números en los que el resultado es "NO" obtendremos la lista de los que no se pueden descomponer de esta forma. Sería la complementaria de la anterior. Podríamos rotular estos números como de categoría 0, ya que no admiten ninguna representación cíclica de tres primos, y a los demás les podemos asignar la categoría según el número de representaciones. Así tendríamos estas categorías:

Categoría 0: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, (faltaría el 31) 32, 33,…

Categoría 1: 31, 41, 59, 61, 87, 91, 103, 113, 119, 121, 129, 143, 161, 171, …

Categoría 2: 71, 101, 131, 211, 221, 269, 271, 343, 359, 391, 401, 423, 437, 439, 451, 471,…

Los primeros números del resto de categorías son:

3	151
4	191
5	491
6	671
7	887
8	311
9	1151

Para la categoría 10 no existe ningún caso inferior a 25000.

TRIÁNGULOS HERONIANOS

Un triángulo se califica de *heroniano* si las longitudes de sus lados y el valor de su área son números enteros. Es un concepto aritmético, por lo que se supone que esto ocurre con una unidad de medida adecuada. Consecuencia de esto es que también es entero el perímetro. A veces se consideran lados y área racionales, pero aquí nos limitaremos a los enteros positivos.

El nombre que les aplicamos es un recuerdo de Herón de Alejandría, autor de la popular fórmula para el área de un triángulo conocidos los lados a, b y c:

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

En esta fórmula, p es el semiperímetro. Si A y los lados son todos enteros, el triángulo será heroniano. Esto exige que p(p-a)(p-b)(p-c) sea un cuadrado perfecto.
Una primera clase de este tipo de triángulos lo constituyen los pitagóricos, ya que en una terna pitagórica (a, b, c), el área equivale a **a*b/2**, y en estas ternas siempre existe un cateto par

(Ver
https://es.wikipedia.org/wiki/Terna_pitag%C3%B3rica)

Por tanto, también será heroniano el formado adosando dos triángulos de este tipo para formar un triángulo isósceles. Así, (20, 26, 26) será de este tipo, porque la altura sobre el lado 20 sería 24, y el área 240.

Igualmente, se pueden adosar dos triángulos pitagóricos que tengan un cateto común, como (5, 12, 13) y (9, 12, 15), formando el triángulo (9+5, 13, 15), es decir, ordenando, (13, 14, 15), que sería heroniano.

Es evidente que, en una terna de lados en un triángulo heroniano, si los multiplicamos todos por un mismo factor, resultará otro heroniano, luego el número de estos será infinito. Llamaremos terna primitiva a aquella en la que los tres lados sean primos entre sí.

Hay ternas que no son pitagóricas, ni tampoco resultado de adosar rectángulos, como (5, 29, 30). Estas reciben el nombre de indescomponibles.

Búsqueda de ternas heronianas

Para encontrar este tipo de ternas existen varios algoritmos, algunos muy eficientes. Nosotros aquí, ya que usamos hojas de cálculo, recurriremos a una rutina que escriba cada terna que encontremos en una columna de la hoja. Como existen infinitas ternas, buscaremos entre dos valores. En contra de nuestra costumbre, usaremos una rutina (macro) en lugar de una función, para escribir los resultados directamente, sin esperar a que la función se evalúe. Lo efectuaremos así:

En las celdas A1 y A2 de la hoja de cálculo escribiremos dos extremos de un intervalo. Lo recorreremos asignando sus valores al lado mayor, y buscaremos una pareja de lados que con él forme un triángulo heroniano. Para ello buscaremos que p(p-a)(p-b)(p-c) sea cuadrado entero. También habrá que tener en cuenta la propiedad de los lados de que "uno de ellos es menor que la suma de los otros dos y mayor que su diferencia". Este ha sido el algoritmo elegido:

Sub heroniano() ‘Es rutina y no función. Deberemos ejecutarla como macro.
Dim t1, t2, a, b, c, p, m, fila, t3

t1 = ActiveWorkbook.Sheets(1).Cells(1, 1).Value ‘Se leen los dos extremos t1 y t2 en A1 y A2
t2 = ActiveWorkbook.Sheets(1).Cells(2, 1).Value
fila = 2 ‘Primera fila de resultados

For a = t1 To t2 ‘El lado **a** se mueve entre los dos extremos

For b = 1 To a ‘El lado **b** será menor o igual que **a**

If a = b Then t3 = 1 Else t3 = a - b + 1 ‘Se busca el inicio para el lado **c**

For c = t3 To b

p = (a + b + c) / 2 ‘Se calcula el semiperímetro

m = p * (p - a) * (p - b) * (p - c) ‘Cuadrado del área (fórmula de Herón)

If escuad(m) Then ‘Si es un cuadrado, es válida

fila = fila + 1 ‘Como hay solución, la fila avanza

ActiveWorkbook.Sheets(1).Cells(fila, 1).Value = a ‘Se escriben a, b, c y el área

ActiveWorkbook.Sheets(1).Cells(fila, 2).Value = b

ActiveWorkbook.Sheets(1).Cells(fila, 3).Value = c

ActiveWorkbook.Sheets(1).Cells(fila, 4).Value = Sqr(m)

End If

Next c

Next b

Next a

End Sub

En la imagen puedes consultar las ternas heronianas en las que el lado mayor está comprendido entre 20 y 25:

	A	B	C	D
1	20			
2	25			
3	20	13	11	66
4	20	15	7	42
5	20	16	12	96
6	21	17	10	84
7	21	20	13	126
8	24	13	13	60
9	24	15	15	108
10	24	20	20	192
11	25	17	12	90
12	25	20	15	150
13	25	24	7	84
14	25	25	14	168
15				

Junto a cada terna figura el área entera. Observamos la abundancia de ternas de este tipo, algo que al iniciar el estudio no se esperaba.

En la página correspondiente de Wikipedia puedes encontrar gran número de propiedades de estos triángulos. Nosotros seguiremos con búsquedas.

Búsqueda de lados conociendo el perímetro

Para esta cuestión volveremos a nuestra estrategia frecuente de comenzar con una función. Dado un número N, devolverá el primer triángulo heroniano que encuentre cuyo perímetro coincida con N. Su desarrollo es similar a la rutina de más arriba.

```
Function pheroniano$(n)
Dim b, c, p, m, a
Dim s$
Dim es As Boolean


s$ = "": es = False
a = n – 2 ‘Comenzamos con el lado mayor a
While a > 1 And Not es
b = 1
While b <= a And Not es
c = n - b - a
If c < a + b And c > a - b Then ‘c es menor que la suma de los otros dos y mayor que su diferencia
p = n / 2
m = p * (p - a) * (p - b) * (p - c)
If escuad(m) Then es = True: s$ = s$ + Str$(a) + Str$(b) + Str$(c) + Str$(Sqr(m))
End If
b = b + 1
Wend
a = a - 1
Wend
If s = "" Then s = "NO"
pheroniano = s
End Function
```

Con esta función es fácil encontrar los primeros perímetros de triángulos heronianos, junto con sus lados y el área:

Perímetro	Lados y área
12	5 3 4 6
16	6 5 5 12
18	8 5 5 12
24	10 6 8 24
30	13 5 12 30
32	15 4 13 24
36	17 9 10 36
40	17 8 15 60
42	20 7 15 42
44	20 11 13 66
48	21 10 17 84
50	24 13 13 60
54	26 3 25 36

Por ejemplo, si el perímetro es 54, el semiperímetro será 27, y el producto 27*(27-26)(27-3)(27-25)=27*1*24*2=1296, cuadrado perfecto, y el área, 36, es entera.

Filtros

En la función anterior podemos introducir filtros para encontrar ternas que cumplan algunas propiedades. Omitimos su código porque no es difícil de reproducir.

Ternas primitivas

Si exigimos que los tres lados a, b y c sean primos entre sí, nos resultarán las ternas primitivas. Estas son las primeras:

12	5 3 4 6
16	6 5 5 12
18	8 5 5 12
30	13 5 12 30
32	15 4 13 24
36	17 9 10 36
40	17 8 15 60
42	15 13 14 84
44	20 11 13 66
48	21 10 17 84
50	24 13 13 60
54	26 3 25 36
56	25 7 24 84
60	29 6 25 60
64	30 17 17 120

Como en casos anteriores, la primera columna es el perímetro, y en la segunda figuran, por este orden, los tres lados y el área. Mentalmente se puede comprobar que los lados son primos entre sí.
Si comparamos con la tabla anterior, vemos, por ejemplo, que falta el 24, porque su terna 10, 6, 8 está formada por los dobles de 5, 4 y 3.

También es destacable que entre estos triángulos aparecen isósceles, como 6, 5, 5 o 24, 13, 13

Con ellos seguimos:

Triángulos isósceles

Por motivos obvios, un triángulo heroniano equilátero no puede existir, porque si el lado es entero, la altura es irracional, pero sí puede ser isósceles, como sería el resultado de adosar dos pitagóricos iguales.

En la búsqueda basta exigir como filtro que a=b o b=c o a=c. Vemos el resultado:

16	6 5 5 12
18	8 5 5 12
32	12 10 10 48
36	16 10 10 48
48	18 15 15 108
50	24 13 13 60
54	24 15 15 108
64	30 17 17 120
72	32 20 20 192
80	30 25 25 300
90	40 25 25 300
96	36 30 30 432
98	48 25 25 168
100	48 26 26 240

Aquí coinciden primitivas y no primitivas. En todas ellas uno de los lados iguales junto con la mitad del desigual formarán un triángulo pitagórico al adjuntarles la altura correspondiente al desigual. Por ejemplo. En 30, 17, 17, el lado 17 y la mitad del 30 forman la terna pitagórica (8, 15, 17)

Lados en progresión aritmética

Existen muchos ejemplos de ternas heronianas con lados en progresión aritmética, pues si son así los de una terna primitiva, lo serán los que son múltiplos de ellos. Por eso es preferible buscar solo entre las primitivas. Para eso, además de exigir que a, b y c sean primos entre sí, exigiremos que formen progresión, es decir, que la diferencia entre dos de ellos coincida con la formada por el tercero. Como no se tiene seguridad de si b>c o c>b, esta condición será doble.

Al implementarlo resultan estos primeros casos:

12	5 3 4 6
42	15 13 14 84
78	37 15 26 156
84	41 15 28 126
114	51 25 38 456
156	75 29 52 546
186	85 39 62 1116
222	87 61 74 2220
228	113 39 76 570
258	97 75 86 3096
294	145 51 98 1176
366	159 85 122 5124
372	183 65 124 2046
402	195 73 134 3216

Vemos que comienzan con dos muy populares, como son 3, 4 y 5, con área 6, y 13, 14, 15, con área 84. No están ordenados. Según el algoritmo usado, el menor se ha escrito en el centro. Para ser primitivas, aparecen más de lo esperado.

Aquí finalizamos los filtros. No se ha intentado con números primos porque la única posibilidad es que uno de ellos fuera 2, y no existe ningún caso.

Números triángulo

Llamaremos números triangulo N a aquellos que se puedan representar como el producto de tres factores N=a*b*c tales que constituyan las medidas de los lados de un triángulo heroniano.

Por ejemplo, 13520=20*26*26, que puede representar un triángulo isósceles cuya altura mide 24, luego su área será entera, e igual a 24*20/2=240.

Si deseamos usar la fórmula de Herón hallaríamos el semiperímetro:

p=(20+26+26)/2=36

La fórmula de Herón se planteará en este caso como

$$A = \sqrt{36(36-20)(36-26)(36-26)}$$
$$= \sqrt{36 * 16 * 10 * 10} = 6 * 4 * 10 = 240$$

Con vistas a encontrar números triángulo, debemos descomponer N en tres factores a, b y c, y exigir después que sea cuadrada la expresión p(p-a)(p-b)(p-c).

La siguiente función se basa en esa idea. Está desarrollada para Excel y Calc, pero es fácil traducirla a otro lenguaje de programación.

Esta función, aplicada a N, devuelve un "NO" si no es número triángulo o una cadena con los tres lados si lo es.

Function esuntriangulo$(n)
Dim a, b, c, d, m, p
Dim es As Boolean
Dim s$

s = "" 'Variable que devuelve la función
es = False 'Control de final de búsqueda
a = 2 'Primer factor
While a <= n / 2 And Not es
m = n / a
If m = Int(m) Then 'Se ve que **a** es un factor de **n**
b = 2 'Segundo factor
While b <= m / 2 And Not es
c = m / b 'Tercer factor
If c = Int(c) Then 'Si los tres son factores, se sigue

d = (a + b + c) / 2 ‘Cálculo del semiperímetro
p = d * (d - a) * (d - b) * (d - c) ‘Producto de Herón
If escuad(p) and p>0 Then ‘Si el producto es cuadrado, N cumple lo planteado
es = True ‘Se interrumpe la búsqueda
s = s + Str$(a) + Str$(b) + Str$(c) ‘Solución
End If
End If
b = b + 1
Wend
End If
a = a + 1
Wend
If s = "" Then s = "NO"
esuntriangulo = s
End Function

Aplicando esta función a los primeros números naturales obtendrás esta sucesión, que está publicada en http://oeis.org/A218243

60, 150, 200, 480, 780, 1200, 1530, 1600, 1620, 1690, 1950, 2040, 2100, 2730, 2860, 3570, 3840, 4050, 4056, 4200, 4350, 4624, 5100, 5400, 5460, 6240, 7500, 8120, 8250, 8670, 8750, 9600, 10812, 11050, 11900, 12180, 12240, 12800, 12960, 13260, 13520, 13650,…

El penúltimo de la lista es 13520, el que nos ha servido de ejemplo.

Por ejemplo, $60=3*4*5$, terna pitagórica que representa a un triángulo de área 6.

Es evidente que si un número está en la lista, $N=a*b*c$, también estará $N*k^3=ak*bk*ck$. Por tanto, esta sucesión es infinita.

SUMAS DE GOLDBACH, LEMOINE Y OTRAS

La conjetura de Lemoine afirma que todo número impar mayor que 5 se puede expresar como la suma p+2q, donde p y q son números primos. Se ha comprobado para N<10^13, y no se ha demostrado cuando escribo esto.

Esta conjetura es más fuerte que la segunda de Goldbach, que afirma que todo número impar mayor que 5 puede expresarse como suma de tres números primos. Aquí no se exige que dos de los primos sean iguales.

Estas dos conjeturas admiten ampliaciones y variantes. Por ejemplo, podemos exigir que dos de los primos sean gemelos, o bien otras más complicadas que podremos tratar si no se alargan las primeras.

En este apartado estudiaremos las soluciones que presenta cada número impar en estas dos conjeturas.

Sumas de Lemoine

Usaremos una función que cuente o presente todas las sumas del tipo p+2q previstas en la conjetura para un número dado. Comenzaremos presentando las sumas además de contarlas. Para ello usaremos la función:

Function sumlemoine(n)
Dim i, j, m
Dim s$

If n Mod 2 = 0 Then sumlemoine = "NO": Exit Function ‘Si ***n*** es par, salimos
m = 0 ‘Contador de soluciones
For i = 2 To n - 4
If esprimo(i) Then ‘Se recorren los primos
j = (n - i) / 2 ‘Se analiza la posible solución para el segundo primo
If esprimo(j) Then m = m + 1: s$ = s$ + "#" + Str$(i) + "+2 *" + Str$(j)
‘Si ambos son primos, se incrementa el contador **m** y se presentan las sumas
End If
Next i
s = Str$(m) + "--" + s
sumlemoine = s
End Function

Con esta función podemos recorrer un conjunto de números impares y comprobar que todos presentan soluciones del tipo N=p+2q. En la tabla figuran los siguientes a 50

51	7--# 5+2 * 23# 13+2 * 19# 17+2 * 17# 29+2 * 11# 37+2 * 7# 41+2 * 5# 47+2 * 2
53	5--# 7+2 * 23# 19+2 * 17# 31+2 * 11# 43+2 * 5# 47+2 * 3
55	3--# 17+2 * 19# 29+2 * 13# 41+2 * 7
57	7--# 11+2 * 23# 19+2 * 19# 23+2 * 17# 31+2 * 13# 43+2 * 7# 47+2 * 5# 53+2 * 2
59	3--# 13+2 * 23# 37+2 * 11# 53+2 * 3
61	3--# 3+2 * 29# 23+2 * 19# 47+2 * 7

En los valores de la función se lee, en primer lugar, el número de soluciones. Así, vemos que 55 presenta 3 y 57, 7. A continuación se escriben las sumas posibles:

55=17+2*19=29+2*13+41+2*7

Este formato es muy ilustrativo, pero en las estadísticas que vamos a estudiar, es un estorbo. Por eso, iremos modificando el resultado, que una vez será el número de soluciones y, en otras ocasiones, máximo, mínimos o diferencias. Sobre la marcha se irá decidiendo.

Número de sumas de Lemoine

Podemos eliminar en la anterior función toda referencia a la cadena de texto s$ y dejar que devuelva solo el número de soluciones. La tabla anterior quedaría así:

51	7
53	5
55	3
57	7
59	3
61	3

De esta forma simplificada se puede crear una lista con los valores en los primeros números impares:

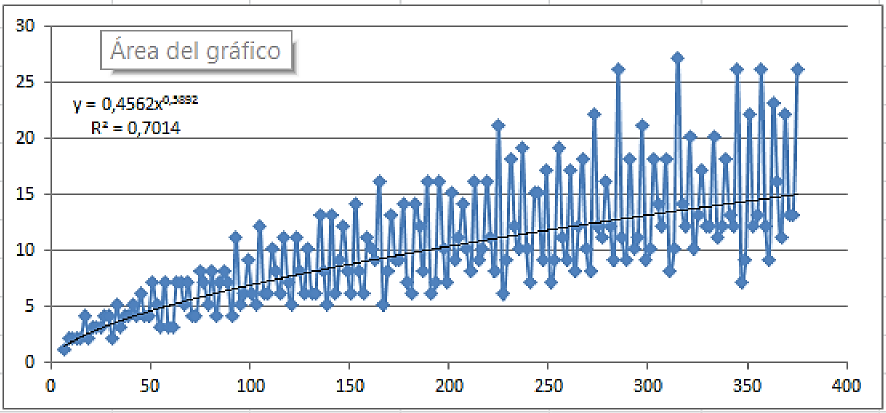
1	0
3	0
5	0
7	1
9	2
11	2
13	2
15	2
17	4
19	2
21	3
23	3
25	3
27	4
29	4
31	2
33	5

Estos valores ya están publicados en http://oeis.org/A046927

A046927 Number of ways to express 2n+1 as p+2q where p and q are primes.

0, 0, 0, 1, 2, 2, 2, 2, 4, 2, 3, 3, 3, 4, 4, 2, 5, 3, 4, 4, 5, 4, 6, 4, 4, 7, 5, 3, 7, 3, 3, 7, 7, 5, 7, 4, 4, 8, 7, 5, 8, 4, 7, 8, 7, 4, 11, 5, 6, 9, 6, 5, 12, 6, 6, 10, 8, 6, 11, 7, 5, 11, 8, 6, 10, 6, 6, 13, 8, 5, 13, 6, 9, 12, 8, 6, 14, 8, 6, 11, 10, 9, 16, 5, 8, 13, 9, 9, 14, 7, 6, 14

Podemos crear un gráfico que compare el valor de cada impar con el número de sumas de Lemoine que presenta:



Observamos que sigue de forma aproximada una tendencia potencial 0,4562x^0,5892, pero con una correlación no muy fuerte, de R^2=0,7014. Esto nos marca una tendencia al crecimiento atenuado en el número de soluciones.

Exploración con CARTESIUS

La obtención de las diversas sumas es un problema combinatorio, y en este tipo de cuestiones puede resultar útil nuestra hoja de cálculo Cartesius

(Descarga desde http://www.hojamat.es/sindecimales/combinatoria/herramientas/herrcomb.htm#cartesius)

Por ejemplo, por las tablas anteriores sabemos que el número 57 admite siete descomposiciones de Lemoine. Lo comprobamos en Cartesius con este planteo:

xtotal=2
xt=1..55
xt=filtro(primo)
ES 2*x1+x2=57

Podemos traducirlo como que

Se usan dos variables
Ambas variarán entre 1 y 55
Se filtran solo los primos
La suma del doble de la primera con la segunda ha de dar 57

El resultado es el previsto, siete posibilidades:

X1	X2	X3
	2	53
	5	47
	7	43
	13	31
	17	23
	19	19
	23	11

El primer primo es el que se multiplica por 2. Así, 2*2+53=57, 2*5+47=57,…

Comparación con las sumas de Golbach

Podemos adaptar la función que hemos presentado al recuento de las soluciones para las sumas de Goldbach para impares, formadas por tres números primos. Tal como se afirmó en los primeros párrafos, se obtendrán valores mayores que en los obtenidos a partir de la conjetura de Lemoine.

Se puede usar la siguiente función:

```
Function sumgoldbach(n)
Dim i, j, m

If n Mod 2 = 0 Then sumgoldbach = 0: Exit Function
m = 0
For i = 2 To n - 4
If esprimo(i) Then
```

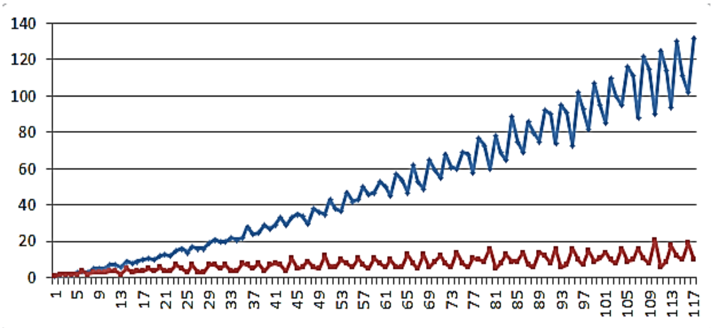
```
j = 2
While j <= i And j <= n - i
If esprimo(j) And esprimo(n - i - j) And j >= n - i - j
Then m = m + 1
j = j + 1
Wend
End If
Next i
sumgoldbach = m
End Function
```

Con ella podemos contar el número de sumas de Goldbach para cada número impar. Están ya publicadas en http://oeis.org/A054860

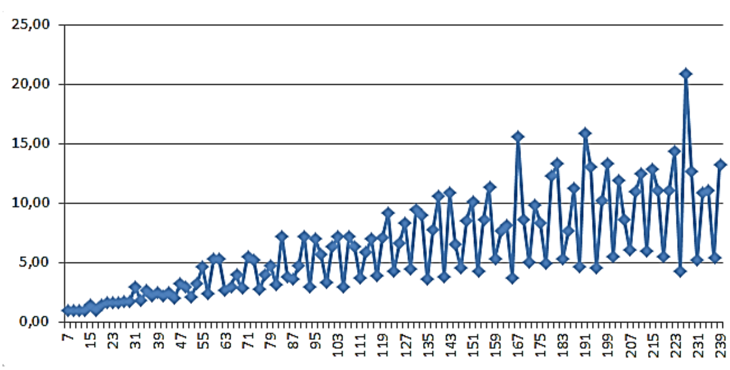
A054860 Number of ways of writing 2n+1 as p + q + r where p, q, r are primes with p <= q <= r.

0, 0, 0, 1, 2, 2, 2, 3, 4, 3, 5, 5, 5, 7, 7, 6, 9, 8, 9, 10, 11, 10, 12, 13, 12, 15, 16, 14, 17, 16, 16, 19, 21, 20, 20, 22, 21, 22, 28, 24, 25, 29, 27, 29, 33, 29, 33, 35, 34, 30, 38, 36, 35, 43, 38, 37, 47, 42, 43, 50, 46, 47, 53, 50, 45, 57, 54, 47, 62, 53, 49, 65, 59, 55,…

Evidentemente, el número de sumas de Lemoine es inferior al de las de Goldbach. En esta gráfica hemos hecho coincidir ambas:



La línea azul sigue las sumas de Goldbach y la roja las de Lemoine. Se observa cómo se ampliando la diferencia entre ellas al crecer los números impares. De hecho, esta es la gráfica de los cocientes de ambas sumas:



En las oscilaciones influyen más las sumas de Goldbach, que son más irregulares en su crecimiento.

Otras sumas de Lemoine

Se han estudiado ya las sumas de Lemoine, en las que los números impares superiores a 5 se descomponen como p+2q, siendo p y q primos. Si 2q lo sustituimos por q-1+q+1, podremos preguntarnos por la posibilidad de que q-1 y q+1 sean los primos (en este caso gemelos), en lugar de q. Es prácticamente el mismo problema, pero más exigente. Existen más números dobles de primos que parejas de primos gemelos. Para estudiar estas sumas bastará modificar ligeramente la función que usamos para las sumas de Lemoine, pero modificando alguna de las líneas del código. Puede ser esta:

```
Function sumlemoine00(n)
Dim i, j, m
Dim s$

s = ""
If n Mod 2 = 0 Then sumlemoine00 = "NO": Exit Function
m = 0
i = 2
While m = 0 And i <= n - 8
If esprimo(i) Then
```

j = (n - i) / 2   ‘Al llegar aquí, se busca un par de primos gemelos

If esprimo(j + 1) And esprimo(j - 1) Then m = m + 1: s$ = s$ + "#" + Str$(i) + "+" + Str$(j - 1) + "+" + Str$(j + 1)

End If ‘El resto del código es muy similar al de las sumas de Lemoine

i = i + 1

Wend

If s = "" Then s = "NO" Else s = Str$(m) + "--" + s

sumlemoine00 = s

End Function

Al aplicar esta función a los primeros impares, no todos presentan una suma de un número primo con un par de primos gemelos:

3	NO
5	NO
7	NO
9	NO
11	1--# 3+ 3+ 5
13	1--# 5+ 3+ 5
15	1--# 3+ 5+ 7
17	1--# 5+ 5+ 7
19	1--# 7+ 5+ 7
21	1--# 13+ 3+ 5

23	1--# 11+ 5+ 7
25	1--# 13+ 5+ 7
27	1--# 3+ 11+ 13
29	1--# 5+ 11+ 13
31	1--# 7+ 11+ 13
33	NO
35	1--# 11+ 11+ 13
37	1--# 13+ 11+ 13
39	1--# 3+ 17+ 19
41	1--# 5+ 17+ 19
43	1--# 7+ 17+ 19
45	1--# 37+ 3+ 5
47	1--# 11+ 17+ 19
49	1--# 13+ 17+ 19
51	1--# 43+ 3+ 5
53	1--# 17+ 17+ 19
55	1--# 19+ 17+ 19
57	NO
59	1--# 23+ 17+ 19
61	1--# 37+ 11+ 13

Vemos que los de una cifra, el 33 y el 57 no admiten ese tipo de suma. De hecho, la gran mayoría de los impares admite la suma p+q+r con p primo y (q,r) par de primos gemelos.

No es fácil encontrar todos los números que no admiten esas sumas. Los primeros son estos:

1, 3, 5, 7, 9, 33, 57, 93, 99, 129, 141, 153, 177, 183, 195, 213, 225, 243, 255, 261, 267, 273, 297, 309, 327, 333, 351, 369, 393, 411, 423, 435, 453, 477, 489, 501, 513, 519, 525, 537, 561, 573, 591, 597, 603, 633, 645, 657, 663, 675, 687, 693, 705, 711, 723, 729, 753, 771, 783, 789, 801, 807, 813, 825,…

Estaban inéditos y los hemos publicado en

https://oeis.org/A329590

Para encontrarlos hemos usado el siguiente código PARI:

for(n = 0, 500, m = 2*n+1; v = 0; forprime(i = 3, m-8, j = (m-i)/2; if(isprime(j-1) && isprime(j+1), v = 1)); if(v == 0, print1(m,", ")))

En él recorremos los impares (m=2*n+1) y después los primos. Para cada primo analizamos si existe un par de primos gemelos en la suma. La variable v recoge el éxito (v=1) o el fracaso (v=0) en la búsqueda. Al final se imprimen los números en los que v=0.

Estudio de un número concreto con CARTESIUS

Con un planteo similar al del anterior tema, podemos encontrar fácilmente las descomposiciones del tipo que estudiamos para un número concreto. Por ejemplo, 61 hemos visto que admite 37+11+13.

Usamos ahora

xtotal=2
xt=1..59
xt=filtro(primo)
ES PRIMO(x1+2)
ES x1+x1+2+x2=61

(La condición ES PRIMO no está implementada en el archivo descargable)

Exigimos que x1+2 sea primo (gemelo con x1), y el resto queda casi igual que en el anterior:

Obtenemos:

X1	X2	X3
3	53	
11	37	

Así que aparece otra solución: 3+5+53=61

Si el número no es muy grande, se puede descomponer con este método. En la imagen vemos las descomposiciones de 121:

X1	X2	X3	X4	X5
3	5	113		121
5	7	109		121
11	13	97		121
29	31	61		121
41	43	37		121

En las cinco soluciones los dos primeros sumandos son primos gemelos.

Pares de primos de Sophie Germain

Por último, podemos exigir que dos de los tres primos de la suma sean un par de Sophie Germain, es decir, que sea primo p y también 2p+1, dejando libre el tercer sumando.

En este caso, están bastante equilibrados el conjunto de los que admiten esta descomposición y los que no:

Los primeros que sí la admiten son estos:

9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 21, 23, 24, 26, 27, 29, 30, 33, 35, 36, 37, 38, 39, 41, 44, 45, 47, 48, 50, 51, 53, 54, 57, 59, 60, 63, 65, 66, 68, 69, 71, 72, 73, 74, 75, 77, 78, 80, 81, 83, 86, 87,…

Por ejemplo, 87 se puede descomponer como:

87=5+11+71=11+23+53=23+47+17

En las tres sumas los dos primeros sumandos son pares de primos de Sofhie Germain.

Los hemos conseguido con Cartesius:

xtotal=2
xt=1..87
xt=filtro(primo)
es primo(2*x1+1)
es x1+2*x1+1+x2=87

Los primeros que no admiten ese tipo de suma son:

1, 2, 3, 4, 5, 6, 7, 8, 11, 16, 22, 25, 28, 31, 32, 34, 40, 42, 43, 46, 49, 52, 55, 56, 58, 61, 62, 64, 67, 70, 76, 79, 82, 84, 85, 88,…

Por ejemplo, estas son las descomposiciones en tres primos del número 43:

3	3	37
3	11	29
3	17	23
5	7	31
5	19	19
7	7	29
7	13	23
7	17	19
11	13	19
13	13	17

En ninguna de ellas aparece una par de primos de Sophie Germain.

Se pueden idear otros condicionamientos con las sumas de Goldbach, pero a ninguna le hemos visto interés. Intenta, por ejemplo, sumas en las que los tres primos formen una progresión aritmética, y llegarás a una trivialidad, y es que coinciden con los triples de los números primos.

SUMAS DE CUADRADOS CON EL MISMO RESULTADO

De nuevo tomamos un tweet de @connumeros para profundizar en una cuestión. El día 1/3/2020 publiqué en Twitter lo siguiente:

1320 es suma de cuadrados pares consecutivos, y también de impares:

Pares: 1320=12^2+14^2+16^2+18^2+20^2=(20×21×22-10×11×12)/6

Impares: 1320=5^2+7^2+9^2+11^2+13^2+15^2+17^2+19^2=(19×20×21-3×4×5)/6

Esto me dio la idea de buscar coincidencias de varias sumas de cuadrados con un mismo resultado. El problema que nos aparecerá será la lentitud de los cálculos, pues nos encontraremos con bucles dobles y triples en los algoritmos de búsqueda. Comenzamos por los más sencillos:

Coincidencias en las sumas de cuadrados

Dado un número natural cualquiera, nos podemos plantear a cuantas sumas de cuadrados equivale. Nos podemos basar en la conocida fórmula de la suma de los primeros cuadrados:

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

Con esta fórmula, si deseamos encontrar una suma **S** de cuadrados que comience en **p** y termine en **q**, su expresión sería S(p,q)=(q(q+1)(2q+1)-(p-1)p(2p-1))/6

Esta expresión se puede implementar en un algoritmo que busque los números que presentan más de dos descomposiciones en suma de cuadrados consecutivos. Lo presentaremos en PARI:

for(n=1, 25000, m=0; i=1; while(i^2<=n, j=0; while(j<i, if(i*(i + 1)*(2*i + 1) - j*(j + 1)*(2*j + 1) == 6*n, m+=1); j+=1); i+=1);if(m>1,print1(n,", ")))

En él, para cada **n** recorremos los valores de **i** mientras **i^2>=n**. Añadimos otra variable **j**, que será el inicio de la posible suma de cuadrados. Usamos la fórmula de más arriba, y si el resultado es **n**, incrementamos el contador **m**. Si este pasa de 1, imprimimos.

Prueba este código en https://pari.math.u-bordeaux.fr/gp.html y obtendrás la siguiente sucesión, que ya está publicada:

A130052 Numbers that are the sum of one or more consecutive squares in more than one way.

25, 365, 841, 1405, 1730, 2030, 3281, 3655, 3740, 4510, 4705, 4760, 4900, 5244, 5434, 5915, 5929, 7230, 7574, 8415, 8464, 9385, 11055, 11236, 11900, 12325, 12524, 14905, 16745, 17484, 18879, 19005, 19044, 19855, 20449, 20510, 21790, 22806, 23681

Aunque la idea de este algoritmo parece acertada, es más rápido este otro, que se limita a sumar cuadrados sin ningún uso de fórmulas. Así que dejamos los dos para comprobar.

```
ok(n) = {my(i=sqrtint(n), m=0, a);while(i>0&&m<2, a=i^2; j=i; while(j>0&&a<=n,if(a==n, m+=1); j-=1; a=a+j^2); i-=1); return(m>1)}

for(p=1, 24000, if(ok(p), print1(p,", ")))
```

Este programa lo hemos añadido a la sucesión publicada.

Coincidencias en sumas de cuadrados impares

La fórmula adecuada para sumar números impares consecutivos es muy parecida a la general:

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

Aunque es útil en otros estudios, parece, tal como se comentó más arriba, que la suma directa de cuadrados es más rápida en lenguaje PARI (y en el VBASIC de Excel) que la suma con esta fórmula. La razón no es que sea ineficiente, sino que requiere bucles de búsqueda más amplios. La usaremos para comprobar.

Para VBasic de Excel usaremos la misma función para cuadrados pares o impares. El listado siguiente sirve para cuadrados impares, y en una de las líneas añadimos como comentario cómo habría que sustituirla para que sirviera para cuadrados pares:

Function vsumacuad3$(n) 'Pares o impares
Dim i, j, m, a
Dim s$

s = ""

i = Int(Sqr(n)) ‘Comenzamos con la posibilidad de un solo cuadrado

If i Mod 2 = 0 Then i = i – 1 ‘Para adaptar al caso PAR, usar ***If i Mod 2 = 1 Then i = i – 1***

m = 0 ‘Número de soluciones

While i > 0 And m < 2

a = i ^ 2

j = i ‘La variable j recorre los posibles sumandos cuadrados

While j > 0 And a <= n

If a = n Then m = m + 1: s = s + "###" + Str$(i) + ", " + Str$(j) ‘Se ha encontrado una suma

j = j – 2 ‘Tanto **j** como **i** bajan de 2 en 2 para mantener la paridad

a = a + j ^ 2

Wend

i = i - 2

Wend

If s = "" Then s = "NO" Else s = Str$(m) + s ‘Se añade el número de soluciones

vsumacuad3 = s

End Function

Buscar soluciones con esta función es una tarea bastante lenta. Como se ha querido llegar a un rango de 2 millones en la búsqueda, ha sido un proceso de muchos minutos. Los primeros números que presentan dos soluciones con sumas de cuadrados impares son estos:

2890, 7735, 22715, 60655, 70225, 87571, 92225, 93314, 136115, 152354, 155519, 256330, 326434, 475861, 511225, 562475, 636360, 671195, 695419, 733485, 808335, 847760, 876490, 1105819, 1107414, 1225965, 1252216, 1293425, 1373701, 1540081, 1541165, 1627899, 1633069, 1832824, 1848405, 1979649

Por ejemplo, 2890 = 37^2 + 39^2 y también 2890 = 7^2 + 9^2 + 11^2 + 13^2 + 15^2 + 17^2 + 19^2 + 21^2 + 23^2 + 25^2

En el lenguaje PARI puedes probar con la fórmula insertada en párrafos anteriores, pero descubrirás pronto su lentitud de proceso. Este sería el código adecuado.

```
for(n=1, 100000, m=0; i=1; while(i^2<=n, j=1; while(j<i, if(i*(i + 1)*(i+2) - j*(j + 1)*(j+2) == 6*n, m+=1); j+=2); i+=2);if(m>1,print1(n,", ")))
```

Después de transcurrir algunos minutos, te devolverá las ocho primeras soluciones: 2890, 7735, 22715, 60655, 70225, 87571, 92225, 93314. Hay que imaginar lo que tardaría en llegar a 2 millones en la búsqueda.

Es más rápido este otro programa:

```
ok(n) = {my(i=sqrtint(n), m=0, a); i=i-(i%2==0); m=0; while(i>0&&m<2, a=i^2; j=i; while(j>0 && a<=n, if(a==n, m+=1); j-=2; a=a+j^2); i-=2); return(m>1)}
concat([0], select(ok, [1..22000]))
```

Suma de cuadrados pares

Las técnicas usadas para los números impares sirven también para los pares, ya que sus fórmulas son similares. En el listado para impares en Vbasic ya lo adveríamos:

If i Mod 2 = 0 Then i = i – 1 'Para adaptar al caso PAR, usar ***If i Mod 2 = 1 Then i = i – 1***

Así obtendríamos:

100, 1460, 3364, 5620, 6920, 8120, 13124, 14620, 14960, 18040, 18820, 19040, 19600, 20976, 21736, 23660, 23716, 28920, 30296, 33660, 33856, 37540, 44220, 44944, 47600, 49300, 50096, 59620, 66980, 69936, 75516, 76020, 76176, 79420, 81796, 82040, 87160, 91224, 94724, 99856

Por ejemplo, 1460=26^2+28^2 y también 1460=20^2+22^2+24^2

Versión en PARI

Prueba este código y obtendrás el mismo listado. Lo hemos organizado sólo hasta 22000 para que no sea tan lento.

ok(n) = {my(i=sqrtint(n), m=0,a,j); i=i-(i%2==1); m=0; while(i>0&&m<2, a=i^2; j=i; while(j>0 && a<=n, if(a==n, m+=1); j-=2; a=a+j^2); i-=2); return(m>1)}

concat([0], select(ok, [1..22000]))

Caso de pares e impares

Llegamos a nuestro objetivo principal, que es buscar aquellos números, como 1320, que admiten sumas de cuadrados pares y también de impares.

Para ello, refundiremos los algoritmos en uno: buscaremos pares e impares por separado y uniremos los resultados mediante una conjunción lógica. En Excel supone un listado bastante largo. Por ello, damos una idea en PARI:

Se idea una función (***issum***) que admita un segundo parámetro además de n (sea t), tal que si vale 0 sirva para los pares y si es 1, para los impares, o al contrario, porque es indiferente. Después, en la función Ok refundimos los dos resultados haciendo una conjunción con la conectiva Y (&& en PARI)

Quedaría así:

```
issum(n,t)={my(i,j,a,m=0);i=sqrtint(n);if(t==0,if(i%2==0,i-=1),if(i%2==1,i-=1));while(i>0&&m<1,a=i^2;j=i;while(j>0&&a<=n,if(a==n,m+=1);j-=2;a=a+j^2);i-=2);return(m)}

isok(m)=issum(m,0)&&issum(m,1)

for(p=1,20000,if(isok(p),print(p)))
```

Con estas dos funciones refundidas obtenemos el listado que pretendíamos. Entre los números encontrados, algunos presentarán soluciones dobles para pares o para impares, pero eso no nos importa por ahora. Lo dejamos por si alguien quiere investigar.

Chocará con la lentitud de los algoritmos.

Los primeros números con esta propiedad mixta son:

164, 596, 1320, 1736, 3156, 4040, 5204, 9416, 10660, 22096, 27080, 29260, 29584, 40020, 69940, 73140, 79540, 85284, 87636, 112916, 113480, 121996, 137960, 161480, 171940, 176420, 182104, 209924, 214396, 221780, 231760, 260120, 290280,…

Por ejemplo, 1736 equivale a estas dos sumas de cuadrados:

Pares: 1736=22^2+24^2+26^2

Impares:

1736=7^2+9^2+11^2+13^2+15^2+17^2+19^2+21^2

BASES DE CUBOS CON SUMA CERO

Otro estudio más que se basa en mis cálculos en Twitter (@connumeros). El día 22/3/2020 publiqué:

22320 se puede representar mediante dos sumas de cubos cuyas bases suman 0:

22320=(-16)^3+(-15)^3+31^3, con 31+(-15)+(-16)=0

22320 =(-60)^3+(-2)^3+62^3 y 62+(-2)+(-60)=0

No son muchos relativamente los números que cumplen una propiedad similar. Comenzaremos con aquellos que presenten suma de cubos cuyas bases sumen cero al menos una vez. El primero es el 6, que se puede representar como 6=2^3+(-1)^3+(-1)^3, con 2+(-1)+(-1)=0

Función adecuada

En primer lugar, hay que destacar que esta condición se puede simplificar. En lugar de usar la igualdad $N=p^3-q^3-r^3$, dado que p=q+r, podemos representar r como p-q.

Según esto, la condición sería $N=p^3-q^3-(p-q)^3$.

Si partimos de esa igualdad, desarrollando, $N=p^3-q^3-(p^3-3p^2q+3pq^2-q^3)=3p^2q-3pq^2=3pq(p-q)$.

Esta expresión 3pq(p-q) nos servirá para construir una parada en la búsqueda, exigiendo que 3pq(p-q)<=N y también para sustituir a $N=p^3-q^3-r^3$. Es más rápido así. También nos indica que N ha de ser múltiplo de 6, ya que pq(p-q) es siempre par.

Versión para Excel

La siguiente función actúa sobre un número natural y devuelve una cadena de texto, que puede estar vacía o contener la primera solución que se encuentre. Este es su listado:

Function cubossum(n)
Dim i, j, a
Dim es
Dim s$

If n Mod 6 <> 0 Then cubossum = "": Exit Function ‘Da salida si no es múltiplo de 6
es = False ‘Parará el proceso si se encuentra solución
i = 1 ‘Contador para la variable ***p***
s = "" ‘Cadena de texto para el resultado
a = 0 ‘Contendrá la suma de cubos
While a <= n And Not es ‘Se para si se llega a ***n*** o se encuentra una suma

j = 1 ‘Contador de la variable ***q***

While j < i And Not es

a = 3 * i * j * (i - j) ‘Expresión buscada

If a = n Then es = True: s = s + Str$(j) + Str$(i) ‘Se encuentra solución

j = j + 1

Wend

i = i + 1

Wend

cubossum = s

End Function

Con esta función podemos organizar una búsqueda de aquellos números que presentan la descomposición buscada. Los primeros son:

6	1 2
18	1 3
36	1 4
48	2 4
60	1 5
90	2 5
126	1 7
144	2 6
162	3 6
168	1 8
210	2 7
216	1 9
252	3 7
270	1 10
288	2 8
330	1 11

Cada número encontrado viene acompañado del valor de q y el de p. Así, para 210, q=2 p=7, luego $210 = 7^3-2^3-(7-2)^3 = 7^3-2^3-5^3 = 343-8-125 = 210$

Un listado más completo es

6, 18, 36, 48, 60, 90, 126, 144, 162, 168, 210, 216, 252, 270, 288, 330, 360, 378, 384, 396, 468, 480, 486, 540, 546, 594, 630, 720, 750, 792, 816, 858, 918, 924, 972, 990, 1008, 1026, 1140, 1152, 1170, 1260, 1296, 1344, 1386, 1404, 1518, 1530, 1560, 1620, 1638, 1656, 1680, 1728, 1800…

Rutina de comprobación

Para comprobar este listado podemos usar técnicas de Excel, ya que permite la creación de listas usando sus flas o columnas. Lo podemos efectuar en tres fases:

1. Se recorren todos los valores posibles de $\mathbf{p^3-q^3-(p-q)^3}$ hasta cierto tope.
2. Se vuelcan todos en la columna A. Aparecerán desordenados y repetidos
3. Se usan los comandos del apartado Datos para ordenar y eliminar duplicados.

En la primera fase usaremos una macro en lugar de una función, para que se puedan situar las soluciones en la columna A. Hemos usado esta:

Sub cuboss()
Dim i, j, k, v

k = 0 ‘Esta variable representa la fila en la que se va a escribir
For i = 1 To 40 *‘Con dos bucles completos se recorren las sumas tipo* p^3-q^3-$(p-q)^3$
For j = 1 To i - 1
k = k + 1 ‘Se incrementa la fila, para escribir en columna
v = i ^ 3 - j ^ 3 - (i - j) ^ 3 ‘Expresión buscada
ActiveWorkbook.Sheets(1).Cells(k, 1).Value = v ‘Se añade a la columna A
Next j
Next i
End Sub

Hemos elegido un tope de 40 en la búsqueda. Para casos sencillos parece excesivo.

Se volcarán los resultados de esta forma:

	A
1	6
2	18
3	18
4	36
5	48
6	36
7	60
8	90
9	90
10	60
11	90
12	144
13	162
14	144
15	90
16	126

Tal como esperábamos, aparecen desordenados y duplicados.

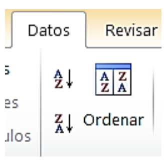
Acudimos a los comandos de Datos:

En primer lugar, seleccionamos la columna A y pedimos **Quitar duplicados**.

Nos indicará que se han quitado miles de duplicados, y quedará:

	A
1	6
2	18
3	36
4	48
5	60
6	90
7	144
8	162
9	126
10	210
11	252
12	168
13	288
14	360
15	384
16	216

Ahora solo queda ordenar:



Así ya lo hemos conseguido, pudiendo comprobar que coincide con el listado anterior.

A
6
18
36
48
60
90
126
144
162
168
210
216
252
270
288
330

Si para un número mayor de términos viéramos que falta alguno, bastaría subir el tope de 40 en la macro.

Todo esto se puede traducir al lenguaje PARI:

```
ok(n) = {my(i=1,a=0,m=0,j);if(n%6==0,while(a<=n&&m==0,j=1;while(j<i&&m==0,a=3*i*j*(i-j);if(a==n,m=1);j+=1);i+=1)); m}

{for(p=1,2000,if(ok(p),print1(p,", ")))}
```

Si lo pruebas en https://pari.math.u-bordeaux.fr/gp.html obtendrás la lista de los primeros números que cumplen esta descomposición:

6, 18, 36, 48, 60, 90, 126, 144, 162, 168, 210, 216, 252, 270, 288, 330, 360, 378, 384, 396, 468, 480, 486, 540, 546, 594, 630, 720, 750, 792, 816, 858, 918, 924, 972, 990, 1008, 1026, 1140, 1152, 1170, 1260, 1296, 1344, 1386, 1404, 1518, 1530, 1560, 1620, 1638, 1656, 1680, 1728, 1800,…

Resultados múltiples

Algunos de estos números presentan varias descomposiciones. El primero es 90, que admite las dos sumas 90=5^3-3^3-2^3 y 90=6^3-5^3-1^3. Después le siguen estos:

90	2 3 5 5 6
630	2 7 10 14 15
720	3 6 10 10 12 15 16
1170	2 10 13 13 15
1260	2 7 12 20 21
1386	2 11 14 21 22
2430	2 9 15 15 18
2640	2 11 16 20 22
3024	2 9 16 14 18
3060	2 12 17 17 20
3168	2 22 24 32 33
3366	2 11 17 33 34
3570	2 10 17 34 35

Si adaptamos a PARI obtenemos un listado más extenso:

90, 630, 720, 1170, 1260, 1386, 2430, 2640, 3024, 3060, 3168, 3366, 3570, 4446, 5040, 5760, 5940, 6210, 6300, 6930, 8910, 9360, 10080, 11088, 11250, 12480, 12870, 12960, 14490, 14742, 16380, 17010, 18018, 18270, 18810, 19440, 19890, 21120, 22140, 22320, 23310, 24192, 24480, 24570, 25344, 25740, 26928, 27360, 27720, 28560, 29700,

Se ha usado el código

```
ok(n) =
{my(i=1,a=0,m=0,j);if(n%6==0,while(a<=n,j=1;while(j
<i,a=3*i*j*(i-j);if(a==n&&j>=i-j,m+=1);j+=1);i+=1));
m>1}
```

{for(p=1,30000,if(ok(p),print1(p,", ")))}

Destaca el 720 con tres descomposiciones:

720=10^3-6^3-4^3=12^3-10^3-2^3=16^3-15^3-1^3

El primero con 4 es 19440: 19440=30^3+(-18)^3+(-12)^3=36^3+(-30)^3+(-6)^3=48^3+(-45)^3+(-3)^3=81^3+(-80)^3+(-1)^3

Con cinco hemos obtenido el 55440, equivale a estas sumas:

55440=42^3+(-22)^3+(-20)^3=44^3+(-30)^3+(-14)^3=55^3+(-48)^3+(-7)^3=70^3+(-66)^3+(-4)^3=80^3+(-77)^3+(-3)^3

Lo dejamos aquí, porque nuestros instrumentos de cálculo se ralentizan con números grandes.

SUMAS CONSECUTIVAS DE CONSECUTIVOS

De nuevo un estudio nuevo se basa en una publicación mía en Twitter (día 7/4/2020)

Iniciamos los cálculos del día 7 con dos sumas de oblongos consecutivos en las que ellas también son consecutivas:

7420=29×30+30×31+31×32+32×33+33×34+34×35+35×36

7420=36×37+37×38+38×39+39×40+40×41

Es muy curiosa esta propiedad, que una suma comience cuando termina otra y que ambas presenten el mismo resultado.

Emprenderemos una búsqueda de sumas que compartan esta propiedad, pero a efectos del algoritmo correspondiente, es preferible dar protagonismo al elemento que separa una suma de otra, en el caso del ejemplo, 35×36. Se puede también basar el cálculo en 36×37, pero había que elegir.

Estructura general el algoritmo

Se puede proponer la siguiente estructura de algoritmo, que servirá, no solo para oblongos, sino para otros tipos de números, como cuadrados o triangulares. Los pasos podrán ser:

- Se construye una función para un valor de **n** de la misma naturaleza que los sumandos. Si no lo es, salimos de la función. En el ejemplo n=35×36. Como es oblongo, seguimos. Si no lo fuera, la función devolvería un valor de error.

- Iniciamos una suma “por la izquierda”. Fijamos **s1 = n**. Definimos una variable **p** con el valor de **n** y un contador **i=1**
- Esa variable **p** descenderá hasta 1, añadiendo sumandos a **s1**
- Para cada valor de **p** y **s1** construimos otra suma **s2** “por la derecha”, a partir de **p+1** sin sobrepasar el valor de **i**.
- Si coinciden **s1** y **s2**, ya hemos terminado
- Si nunca coinciden, devuelve un “NO”

Lo entenderás mejor con el listado de la función para enteros.

Propiedad para enteros

Buscaremos, en primer lugar, aquellas sumas de este tipo formadas por enteros. Por ejemplo, el número 8 es centro de dos sumas consecutivas:

4+5+6+7+8=30=9+10+11

En este ejemplo, i=4, porque hay que tomar 4 sumandos a la izquierda del 8, y j=2, porque se toman 2 a partir de 8+1=9. Se puede cambiar este conteo si se desea. Aquí, s1=s2=30.

La función que resuelve esto puede ser:

Function igualsuma(n) 'con enteros
Dim i, j, p, p1, q, s1, s2
Dim a$

a = ""
s1 = n: p = n: p1 = p: i = 1
While p > 1 ‘i llega hasta 1 como máximo
p = p - 1
s1 = s1 + p ‘Construimos s1
q = p1: s2 = 0: j = 0
While j < i ‘j no sobrepasa a i, porque ha de ser menor
q = q + 1
s2 = s2 + q ‘Construimos s2
If s1 = s2 Then a = a + "&&" + Str$(n) + "//" + Str$(i) + ", " + Str$(j) + " S1 " + Str$(s1) + " S2 " + Str$(s2)
j = j + 1
Wend
i = i + 1
Wend
If a = "" Then a = "NO"
igualsuma = a
End Function

Si emprendemos una búsqueda obtendremos los primeros números que son centro de dos sumas iguales. Algunos, como el 12, presentan dos coincidencias distintas en las sumas:

2 && 2// 1, 0 S1 3 S2 3
6 && 6// 2, 1 S1 15 S2 15
7 && 7// 5, 2 S1 27 S2 27
8 && 8// 4, 2 S1 30 S2 30
12 && 12// 3, 2 S1 42 S2 42&& 12// 9, 4 S1 75 S2 75
14 && 14// 13, 5 S1 105 S2 105
17 && 17// 13, 6 S1 147 S2 147
18 && 18// 6, 4 S1 105 S2 105
19 && 19// 8, 5 S1 135 S2 135
20 && 20// 4, 3 S1 90 S2 90
22 && 22// 17, 8 S1 243 S2 243
25 && 25// 14, 8 S1 270 S2 270
26 && 26// 17, 9 S1 315 S2 315
27 && 27// 21, 10 S1 363 S2 363
30 && 30// 5, 4 S1 165 S2 165

Estos resultados se pueden interpretar de la siguiente forma, que vemos con el ejemplo del 18:

18 && 18// 6, 4 S1 105 S2 105

En primer lugar leemos el centro o pivote de las sumas, en este caso, && 18//. Los siguientes números 6 y 4 son respectivamente los sumandos tomados a la izquierda del 18 y los tomados a la derecha a partir del 19, incluyendo este (serían 5 en total). Estas serían las dos sumas consecutivas:

12+13+14+15+16+17+18=105, que es el valor s1 que devuelve la función

19+20+21+22+23=105, que es el valor devuelto como s2, coincidente con s1.

Observamos que existen muchos números que cumplen esta propiedad, y que incluso se forman grupos de consecutivos. Es normal, por ser números enteros que presentan diferencias pequeñas en sus sumas. Aparecerán menos con otros tipos de números.

Números oblongos

Este fue el caso que publiqué en Twitter. Podemos usar el mismo esquema para enteros, con las siguientes diferencias.

(1) Si el número no es oblongo, salimos de la función con un “NO”

Para saber si un número O es oblongo, hay que recordar que será el doble de un triangular T, y que estos se caracterizan porque 8T+1 es un cuadrado. Así que en los oblongos O será cuadrado 4O+1. Aquí tienes una implementación para Excel y Calc:

Public Function esoblongo(n) As Boolean
If escuad(4 * n + 1) Then esoblongo = True Else esoblongo = False
End Function

(2) Las sumas del listado de arriba, **s1=s1+p** y **s2=s2+q** se cambiarán a **s1=s1+p*(p+1)** y **s2=s2+q*(q+1)**, para que cada sumando sea oblongo.

Con estos cambios, y quizás algún otro menor, obtendremos el listado de los primeros números equivalentes a dos sumas de oblongos consecutivos, que a su vez son consecutivas:

12 && 12// 2, 0 S1 20 S2 20
72 && 72// 3, 1 S1 200 S2 200
240 && 240// 4, 2 S1 920 S2 920
600 && 600// 5, 3 S1 2920 S2 2920
1260 && 1260// 6, 4 S1 7420 S2 7420
2352 && 2352// 7, 5 S1 16240 S2 16240
4032 && 4032// 8, 6 S1 31920 S2 31920
4692 && 4692// 33, 15 S1 95200 S2 95200
5852 && 5852// 69, 19 S1 152040 S2 152040
6480 && 6480// 9, 7 S1 57840 S2 57840
9900 && 9900// 10, 8 S1 98340 S2 98340
10100 && 10100// 77, 25 S1 339352 S2 339352
14520 && 14520// 11, 9 S1 158840 S2 158840
17030 && 17030// 86, 32 S1 720940 S2 720940

20592 && 20592// 12, 10 S1 245960 S2 245960
28392 && 28392// 13, 11 S1 367640 S2 367640

Podemos observar en la quinta fila el caso que publiqué:

1260 && 1260// 6, 4 S1 7420 S2 7420

A partir de 1260=35*36, oblongo por tanto, se toman 6 sumandos más a la izquierda y 4 más a la derecha de 36*37, reproduciéndose así lo que se publicó en su día.

Con primos

Ya está publicado en http://oeis.org/A089930

A089930 Primes p such that there exists a set of consecutive primes ending with p which has the same sum as a set starting right after p.

3, 13, 47, 73, 83, 269, 349, 359, 487, 569, 569, 787, 859, 929, 941, 1171, 1237, 1297, 1307, 1429, 1549, 1553, 1607, 1877, 2011, 2083, 2111, 2113, 2389, 2399, 2557, 2579, 2633, 2659, 2677, 2749, 2777, 2837, 2969, 3001, 3019, 3019, 3067, 3119, 3169, 3203,…

Nos hemos limitado a adaptar el algoritmo inicial al caso de primos. Los resultados concuerdan con los publicados:

3 && 3// 1, 0 S1 5 S2 5&& 3// 2, 0 S1 5 S2 5
13 && 13// 3, 1 S1 36 S2 36
47 && 47// 10, 4 S1 311 S2 311
73 && 73// 9, 5 S1 552 S2 552
83 && 83// 17, 7 S1 846 S2 846
269 && 269// 36, 18 S1 6231 S2 6231
349 && 349// 56, 24 S1 10649 S2 10649
359 && 359// 61, 25 S1 11470 S2 11470
487 && 487// 63, 31 S1 19066 S2 19066
569 && 569// 13, 11 S1 7256 S2 7256&& 569// 71, 35 S1 24518 S2 24518

En cada suma se añaden o los primos anteriores o los posteriores a cada sumando. Observamos, por ejemplo, que 569 equivale a dos casos distintos. Desarrollamos el primero:

&& 569// 13, 11 S1 7256 S2 7256

Deberemos tomar 13 primos anteriores a 569 y consecutivos con él, incluyéndolo en la suma. Después, 11 primos siguientes a 569, también incluyendo ese siguiente. Lo comprobamos:

Decreciente:

569+563+557+547+541+523+521+509+503+499+491+487+479+467=7256

Creciente:

571+577+587+593+599+601+607+613+617+619+631+641=7256

El segundo caso supone sumas de muchos sumandos, y lo dejamos sin comprobar.

Con triangulares y cuadrados

Como el tema ya está bastante estudiado con los ejemplos anteriores, solo añadiremos los primeros números que cumplen la propiedad para números triangulares y para los cuadrados.

6 && 6// 2, 0 S1 10 S2 10
36 && 36// 3, 1 S1 100 S2 100
120 && 120// 4, 2 S1 460 S2 460
300 && 300// 5, 3 S1 1460 S2 1460
630 && 630// 6, 4 S1 3710 S2 3710
1176 && 1176// 7, 5 S1 8120 S2 8120
2016 && 2016// 8, 6 S1 15960 S2 15960
2346 && 2346// 33, 15 S1 47600 S2 47600
2926 && 2926// 69, 19 S1 76020 S2 76020
3240 && 3240// 9, 7 S1 28920 S2 28920
4950 && 4950// 10, 8 S1 49170 S2 49170
5050 && 5050// 77, 25 S1 169676 S2 169676

7260 && 7260// 11, 9 S1 79420 S2 79420
8515 && 8515// 86, 32 S1 360470 S2 360470

Por ejemplo, 1460 equivale a estas dos sumas de triangulares:

1460=190+210+231+253+276+300

1460=325+351+378+406

Todos los triangulares de estas dos sumas son consecutivos, desde 190=19*20/2, hasta

406 =28*29/2

Cuadrados

Los primeros casos son:

16   && 16// 1, 0 S1 25 S2 25
144   && 144// 2, 1 S1 365 S2 365
576   && 576// 3, 2 S1 2030 S2 2030
1156 && 1156// 16, 7 S1 11900 S2 11900
1444 && 1444// 34, 9 S1 19005 S2 19005
1600 && 1600// 4, 3 S1 7230 S2 7230
2500 && 2500// 38, 12 S1 42419 S2 42419
3600 && 3600// 5, 4 S1 19855 S2 19855
7056 && 7056// 6, 5 S1 45955 S2 45955
12100 && 12100// 50, 24 S1 379525 S2 379525
12544 && 12544// 7, 6 S1 94220 S2 94220
20164 && 20164// 126, 36 S1 963295 S2 963295

20736 && 20736// 8, 7 S1 176460 S2 176460
25281 && 25281// 92, 38 S1 1254539 S2 1254539
32400 && 32400// 9, 8 S1 308085 S2 308085
48400 && 48400// 10, 9 S1 508585 S2 508585
69696 && 69696// 11, 10 S1 802010 S2 802010
97344 && 97344// 12, 11 S1 1217450 S2 1217450

Un ejemplo sencillo es el de 1600 como separador de las dos sumas:

1600=40^2, y se tiene:

36^2+37^2+38^2+39^2+40^2=7230

41^2+42^2+43^2+44^2=7230

Ambas sumas coinciden, luego se cumple la propiedad pedida.

Con estos ejemplos vemos que la propiedad es exigente, y pocos números la cumplen, pero yo esperaba que aparecieran menos.