Otro algoritmo de números primos

La conjetura de GoldBach es tan antigua e insondable que dada su sencillez nos produce asombro:

"cualquier número par se puede expresar como suma de dos números primos"

Esto nos lleva a pensar que si descomponemos un número par en sumas de números primos (con respecto al número par) algunas de ellas serán de números primos entre sí (necesariamente según la conjetura apuntada).

Es fácil demostrar (no lo haré aquí) que si el número (par) 2*n+2 = p + q, y p y q son primos entre sí, ni p ni q dividen al número. Esto se cumplirá claramente en el caso de que p y q sean primos.

Como la conjetura nos asegura que habrá al menos un par de primos, sólo para el caso de que p=q (por ejemplo 6 = 3+3, o 10=5+5, etc.), p y q tienen que ser necesariamente distintos.

Esto nos anima a que encontremos un algoritmo que dado un número par, genere las distintas sumas p + q de manera que ni p ni q sean factores del número par.

En otras palabras, buscamos la descomposición en no-factores primos de un número par.

Así, la descomposición en no-factores primos de 14 es 3+11, 5+9 y la de 54 es 5+49, 7+47, 11+43, 13+41, 17+37, 19+35, 23+31, 25+29.

Encontramos que un número par 2*n+2>= 14 puede descomponerse en x sumas de no-factores primos, de manera que en todas ellas se cumple que los no-factores son primos entre sí y con respecto a 2*n+2.

En no todas las sumas los dos no-factores son primos (sí serán primos entre sí y con respecto a la suma).

Lo que hace el algoritmo que proponemos es contar la frecuencia de aparición del no-factor primo segundo de cada suma, partiendo de una lista mínima de primos: 3,5 y 7 (el 2 no interesa para nuestros cálculos) con frecuencias de aparición 0,1 y 2, respectivamente, ya que el 3 aparece en el 8 pero en primer lugar, el 5 aparece en el 8 en segundo lugar y el 7 en el 10 y en el 12 también en segundo lugar.

La siguiente gráfica da dicha frecuencia de aparición para los números inferiores a 50 (figura 1).

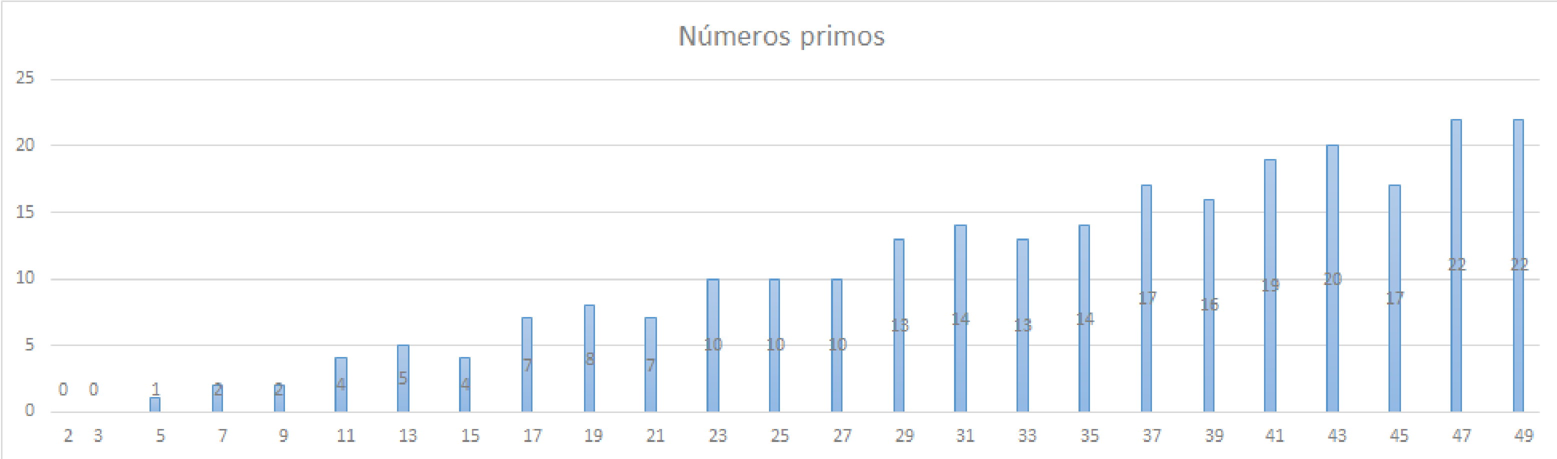

Figura1. Posibles primos entre 7 y 50

Como vemos están todos los números impares (y por lo tanto primos) pero hay muchos de ellos que no son primos (como era de esperar).

Si sólo nos quedamos con los números en los que la frecuencia no decrece (Figura 2).

Con lo que encontramos todos los números primos entre 2 y 50 con la excepción del 35 y el 49 que son compuestos.

Podemos eliminar el 35 si consideramos que la pendiente de la curva sea positiva (Figura3).

Ahora son todos los que son están.

Si ahora tomamos los primos menores que 100 (Figura 4).

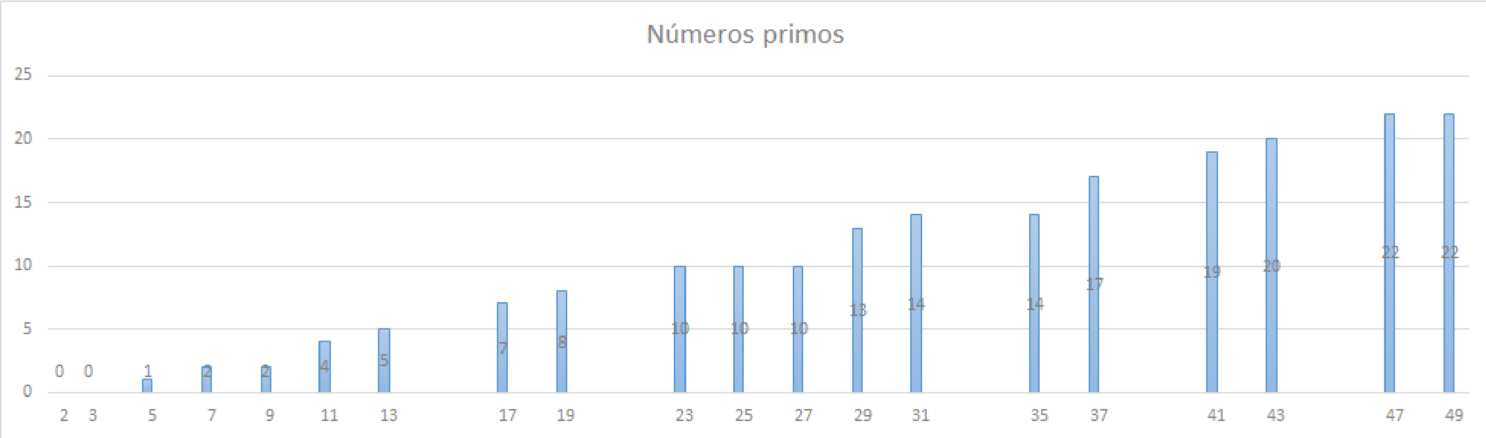

Figura2. Posibles primos entre 7 y 50 mejorada.

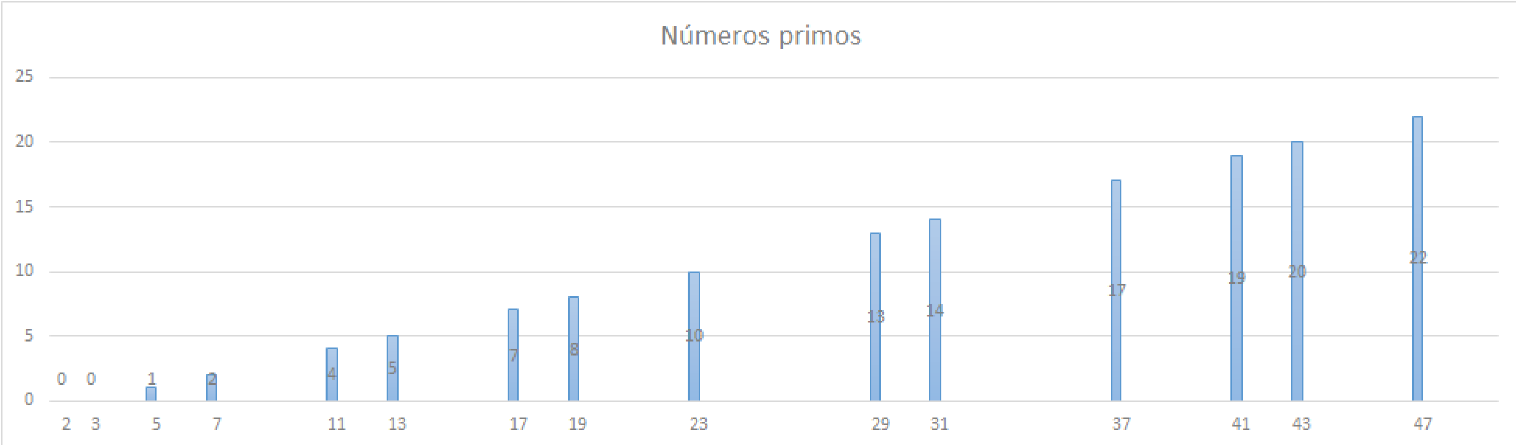

Figura3. Todos los que son y están.

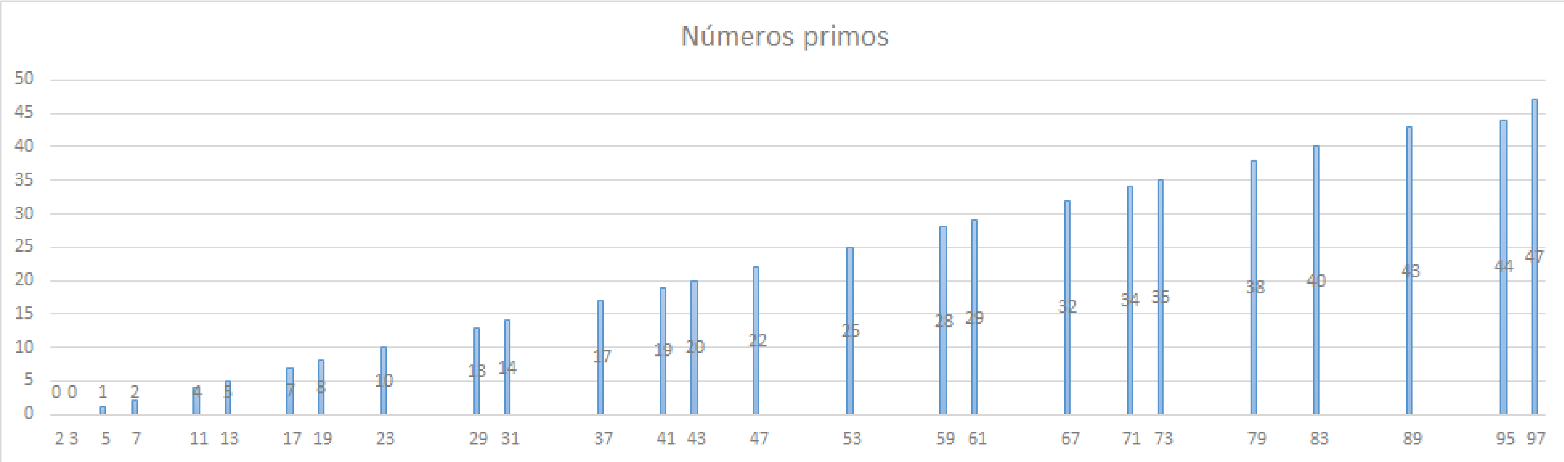

Figura 4. Primos entre 7 y 100

Nos sobra el 95.

Podemos eliminar el 95 al considerar que está muy separado del anterior (89) y ha crecido poco (1), es decir que la pendiente es “pequeña” (Figura 5).

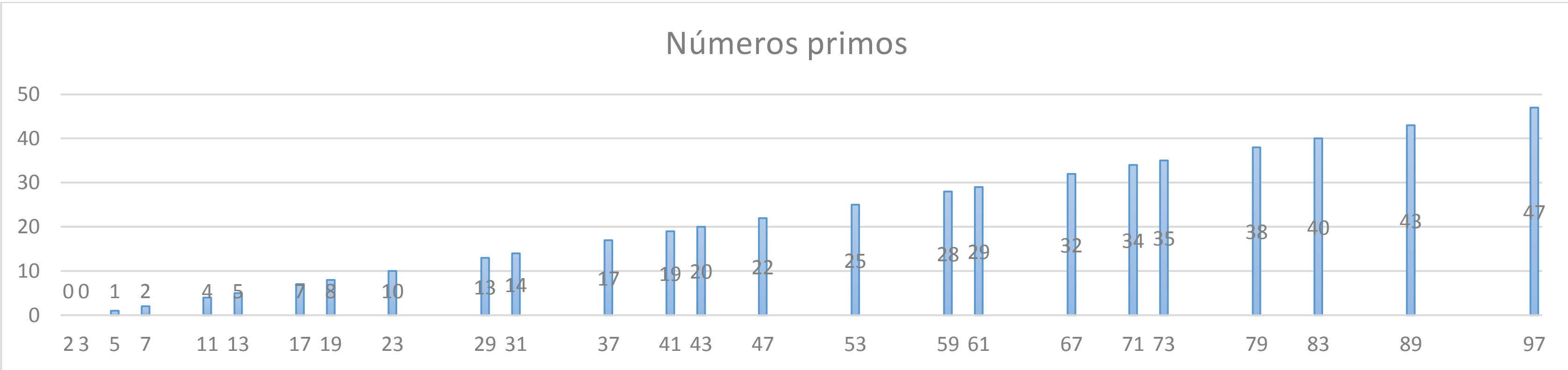

Figura 5. Última mejora con los primos menores que 100

En concreto, la pendiente para los primos resulta ser de 0,5. La función f(x) con f la frecuencia de aparición de un primo en el segundo sumando de la descomposición en no-factores primos de un par es f(x)=0,5*x -1,5; con x un número primo.

Si x es impar pero no primo f(x) no coincidirá con 0,5*x -1,5.

Los primos entre 97 y 200 resultan (figura 6).

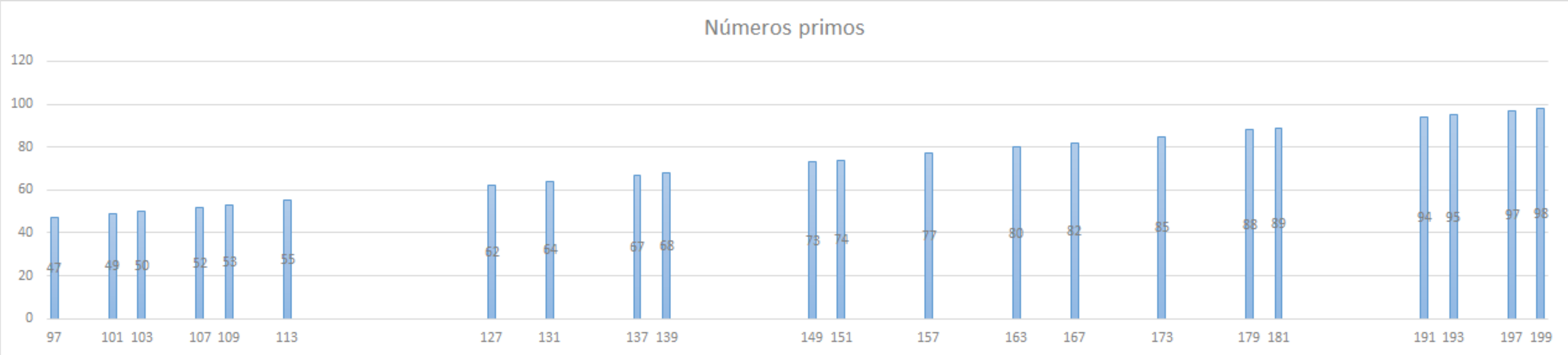

Figura 6. Primos entre 97 y 200

¡Hurra!

Aunque el algoritmo es lento (no tiene por qué ser rápido) nos determina exactamente los primos en la secuencia de números impares.