Certificación de la constante singular de OEIS A379427

Enunciado

Sean $L_i(n) = a_i n + b_i$ con
$(a_i, b_i) = (2,3), (3,5), (5,7), (7,11), (11,13), (13,17), (17,19), (19,23)$, y
$\nu(q) = \#\{r \bmod q : \ q \mid \prod_i L_i(r)\}$. Entonces

$$\mathfrak{S} = \prod_q \frac{1 - \nu(q)/q}{(1-1/q)^8} \in 1440.500641588221780972612428946904712298351522344822417318639$$

Se certifican al menos 230 cifras decimales correctas (en el artículo bastan, por ejemplo, 50):

$$\mathfrak{S} = 1440.500641588221780972612428946904712298351522344 82\ldots$$

Lema 1: primos excepcionales

Si $q \nmid a_i$ para todo i, cada L_i tiene una única raíz $r_i \equiv -b_i a_i^{-1} \pmod q$, y $r_i \equiv r_j$ si y solo si $q \mid \Delta_{ij} = a_i b_j - a_j b_i$. Los 28 determinantes son no nulos y todos sus factores primos, junto con los de los a_i, están en $\{2,3,5,7,11,13,17,19\}$ (comprobado por el programa). Por tanto $\nu(q) = 8$ para todo primo $q \geq 23$. Para $q \leq 19$: $\nu = 1,2,4,4,6,5,7,7$.

Lema 2: expansión logarítmica

Para $q > Q = 1000$, con $\nu(q) = 8 < q$,

$$log frac 1 - 8/q (1-1/q)^8 = - sum_k ge 2 frac 8^k - 8k, q^k$$

(el término $k = 1$ se anula). Todos los términos son del mismo signo y la serie doble converge absolutamente, así que

$$log mathfrak S = sum_q le Q log frac q - nu(q) q - 8 log frac q - 1 q; - ; sum_k ge 2 frac 8^k - 8k P_Q(k$$

Lema 3: zeta prima sin los primos pequeños

Sea $\zeta_Q(s) = \zeta(s) \prod_{p \leq Q}(1 - p^{-s}) = \prod_{p > Q}(1-p^{-s})^{-1}$. Por inversión de Möbius (fórmula clásica de la zeta prima, aplicada al producto de Euler truncado)

$$P_Q(k) = sum_m ge 1 frac mu(m) m log zeta_Q(mk).$$

Lema 4: cotas de los restos

(a) Para $s \geq 2$: $0 \leq \zeta_Q(s) - 1 \leq \sum_{n>Q} n^{-s} \leq Q^{1-s}/(s-1) \leq \frac{1}{2}$, y como $\log(1+x) \leq x$, $0 \leq \log \zeta_Q(s) \leq Q^{1-s}$. Por tanto, al truncar la serie de Möbius en $m < M$:

$$Bigl|sum_mgeMtfracmu(m)mlogzeta_Q(mk)Bigr|lesum_mgeMQ^{1-mk} = fracQcdotQ^{-Mk}1 -$$

(el programa usa el doble de esta cota, aún más holgado).

(b) Como $0 \leq P_Q(k) \leq Q^{1-k}/(k-1)$, el resto en $k > K$ cumple

$$0lesum_k > Kfrac8^k - 8kP_Q(k)lesum_k > Kfrac8^kQ^{1-k}k(k-1)lefrac8^{K+1}Q^{-K}K(K+1)ca$$

Con $Q = 1000$, $K = 143$ esta cota es menor que 10^{-297}.

Control de los errores numéricos

Todos los cálculos se hacen con aritmética de bolas Arb (python-flint): cada número es un intervalo $[m \pm r]$ cuyo radio incluye, de forma rigurosa, todos los errores de redondeo, y $\zeta(s)$, log y exp devuelven bolas garantizadas que contienen el valor exacto. Las cotas de los Lemas 4(a) y 4(b) se añaden como bolas $[0 \pm \varepsilon]$. La bola final contiene, por tanto, con certeza el valor exacto de $\mathfrak{S}$. A 1200 bits el radio es 1.21×10^{-233}. Con 400 bits se obtiene un radio de 9.1×10^{-73} y valores consistentes (control independiente).

Comparación

Cantidad	Valor	$\mathfrak{S}-$ valor
Aproximación previa con la zeta prima	1440.5006415882217809726124	$+2.8947 \times 10^{-23}$
Producto truncado hasta $2 \cdot 10^6$	1440.501946420289	-1.30483×10^{-3}

Las 25 cifras de la aproximación previa son correctas: la diferencia es solo el redondeo en la última cifra (el valor certificado continúa así: ...809726124289...). El producto truncado excede al límite en 0.0013, lo que concuerda con la cola $\approx \exp(-28/(N \log N))$ con $N = 2 \cdot 10^6$.

Reproducción

```
pip install python-flint sympy
python3 certify_S.py 1200   # el argumento es la precisión en bits
```

El programa (`certify_S.py`) imprime $\nu(q)$, los primos excepcionales, $\log \mathfrak{S}$ y $\mathfrak{S}$ como bolas certificadas, y las diferencias con los dos valores anteriores. Tarda alrededor de 1 segundo.