Cálculo del número exacto de primos menores que x. El método de Mapes.

Philip Pita Forrier
Trabajo de fin de grado de Matemáticas
Universidad de Zaragoza

Directores del trabajo:
Julio Bernués Pardo
Ángel R. Francés Román
26 de junio de 2022

Abstract

The aim of this paper is to study the exact number of primes less than or equal to a given x. Such a quantity is usually denoted by $\pi(x)$; that is,

$$\pi(x) = \#\{p \in \mathbb{N};\ p \text{ is prime and } p \leq x\}$$

In Chapter 1 we study some elementary theorems on the distribution of primes. First, the Prime Number Theorem is stated as follows:

$$\frac{\pi(x)}{x/\log x} \longrightarrow 1\ , \quad \text{as } x \to \infty\ .$$

Then, the Chebyshev functions ψ and ϑ are defined, showing that they allow us to express an alternative formulation of the Prime Number Theorem, since the following relations are equivalent:

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1\ , \qquad \lim_{x\to\infty} \frac{\vartheta(x)}{x} = 1\ , \qquad \lim_{x\to\infty} \frac{\psi(x)}{x} = 1\ .$$

Further on, considering the series of the reciprocals of the primes $\sum_p 1/p$, the following asymptotic expression is found for its partial sums:

$$\sum_{p\leq x} \frac{1}{p} = \log\log x + A + O\left(\frac{1}{\log x}\right), \quad \forall x \geq 2\ ,$$

where A is a certain constant. This not only proves that the aforementioned series diverges, a fact that had already been proved by Euler, but also reinforces the Theorem proved by Euclid on the existence of infinite primes.

The purpose of Chapter 2 is to construct Mapes' method for computing $\pi(x)$. Such a method requires knowing a certain number of primes. The oldest of these methods is the sieve of Eratosthenes, which obtains all prime numbers less than or equal to x, so that they can be counted.

Legendre's formula is introduced, given by

$$\pi(x) = a - 1 + \phi(x,a)\ ,$$

where $a = \pi(\sqrt{x})$ and the quantity $\phi(x,a)$ counts the positive integers less than or equal to x that are not divisible by any of the first a primes. For this last value, called the Legendre sum, the following explicit formula will be derived:

$$\phi(x,a) = x - \sum_{1\leq i\leq a} \left\lfloor \frac{x}{p_i} \right\rfloor + \sum_{1\leq i<j\leq a} \left\lfloor \frac{x}{p_i p_j} \right\rfloor - \sum_{1\leq i<j<k\leq a} \left\lfloor \frac{x}{p_i p_j p_k} \right\rfloor + \cdots + (-1)^a \left\lfloor \frac{x}{p_1 p_2 \cdots p_a} \right\rfloor\ .$$

By defining the values $P_k(x,a)$, which count the positive integers less than or equal to x that can be expressed as a product of k primes greater than p_a, we prove that they define a rearrangement of the terms of the Legendre sum. On this basis, Meissel and Lehmer's methods are constructed, which improve Legendre's method by reducing the value of a, thus speeding up the calculation of $\phi(x,a)$ and hence that of $\pi(x)$.

We state a set of properties that show the recusive character of the function $\phi(x,a)$. By creating certain tables where some values of the function are stored, these properties will allow us to exploit the nature of $\phi(x,a)$ to compute it for arbitrarily large values of x for all a less than a certain constant. In addition, a formula derived from Lehmer's method will be given which allows us to compute $\phi(x,a)$ very efficiently whenever a sufficiently large number of primes is available.

Lastly, Mapes' method is introduced, starting from the definition of the values

$$T_k(x,a) = (-1)^{\beta_0+\beta_1+\cdots+\beta_{a-1}} \left\lfloor \frac{x}{p_1^{\beta_0} p_2^{\beta_1} \cdots p_a^{\beta_{a-1}}} \right\rfloor ,$$

where $k = 2^{a-1}\beta_{a-1} + 2^{a-2}\beta_{a-2} + \cdots + 2^1\beta_1 + 2^0\beta_0$, being $\beta_i \in \{0,1\}\ \forall i$. These values will univocally assign each term of $\phi(x,a)$ to a certain index k, allowing to form groupings of the terms of $\phi(x,a)$, called $\gamma(M,x,a)$, that will be used to decompose $\phi(x,a)$ into successively smaller terms by means of the identity

$$\phi(T_M(x,a),i) = T_M(x,a) + \phi(T_{M+2^0}(x,a),0) + \phi(T_{M+2^1}(x,a),1) + \cdots + \phi(T_{M+2^{i-1}}(x,a),i-1)$$

until they can be calculated exclusively using either the ϕ tables or Lehmer's formula mentioned above.

In addition, we have implemented this method in Java and it has been used to obtain a table of primes.

Índice general

Abstract **iii**

1. Sobre la Distribución de los Números Primos **1**
1.1. Formas equivalentes del Teorema de los Números Primos 2
1.2. Expresión asintótica de las sumas parciales de la serie de los recíprocos de los primos . . 5

2. Método de Mapes **11**
2.1. La criba de Eratóstenes 11
2.2. Fórmula de Legendre 13
2.3. Métodos de Meissel y Lehmer 14
2.4. Cálculo de la suma de Legendre 17
2.5. Método de Mapes 20

Bibliografía **25**

Apéndices **27**
A. Tabla de Primos 27
B. Código del Método de Mapes 29
B.1. Clase *Phi* 29
B.2. Clase *BigInt* 31
B.3. Clase *Mapes* 33

Capítulo 1

Sobre la Distribución de los Números Primos

Los números primos y sus propiedades han atraído la atención de los matemáticos desde la antigüedad. Por ejemplo, en el séptimo libro de los Elementos de Euclides (300 A.C.) se hallan las definiciones (Definición 11) de números primos y compuestos, se demuestra (Proposición 30) que si un número primo divide a un producto, entonces debe dividir a alguno de sus factores y también se demuestra en el noveno libro (Proposición 20) la existencia de infinitos números primos.

Nuestro objetivo en este trabajo se centra en calcular la cantidad exacta de números primos que son menores o iguales a cierto $x \in \mathbb{R}^+$. En la literatura se denota habitualmente $\pi(x)$ a esta cantidad; esto es,

$$\pi(x) = \#\{p \in \mathbb{N};\ p \text{ es primo y } p \leq x\} \ .$$

Gauss [6] (1791) señaló que la cantidad $\pi(x)$ debería crecer aproximadamente como $x/\log x$. Pocos años más tarde, Legendre [9] (1798) conjeturó que $x/(A \log x + B)$ debería aproximar bien a $\pi(x)$, con A y B constantes.

Durante más de un siglo, las conjeturas sobre el valor asintótico de $\pi(x)$ permanecieron sin demostración. El primer resultado destacable llegó de parte de Chebyshev [3] (1848), quién demostró exitosamente que si el límite siguiente existe, debe ser igual a -1:

$$\lim_{x\to\infty} \left(\frac{x}{\pi(x)} - \log x \right) \ .$$

El desarrollo de la teoría de funciones complejas supuso un importante impulso en el estudio del valor asintótico de $\pi(x)$. En concreto, Riemann [15] (1860) estudió por primera vez las propiedades de la función

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

para valores complejos de s, la cual se relaciona con los números primos mediante la siguiente identidad, hallada por Euler [5] (1737) para $s > 1$ real:

$$\zeta(s) = \prod_p \frac{1}{1 - p^{-s}} \ .$$

La contribución de Riemann permitió que pocos años más tarde se culminara el estudio del valor asintótico de $\pi(x)$ mediante la demostración de Hadamard [7] (1896) y de la Vallée-Poussin [17] (1896) del Teorema de los Números Primos, que afirma que

$$\frac{\pi(x)}{x/\log x} \longrightarrow 1 \ , \quad \text{cuando } x \to \infty \ .$$

1.1. Formas equivalentes del Teorema de los Números Primos

A continuación se describirán y demostrarán dos formulaciones equivalentes al Teorema de los Números Primos, dadas a partir de las funciones ψ y ϑ de Chebyshev, cuya definición es la siguiente.

Definición 1. Para todo entero $n \geq 1$ se define

$$\Lambda(n) = \begin{cases} \log p & , \text{ si } n = p^m \text{ para algún primo } p \text{ y } m \geq 1 \\ 0 & , \text{ en otro caso} \end{cases}$$

Además, para $x > 0$ se define la función ψ de Chebyshev dada por

$$\psi(x) = \sum_{n \leq x} \Lambda(n) .$$

Observación. Al ser $\Lambda(n) = 0$ salvo cuando n es la potencia de un primo, $\psi(x)$ puede expresarse como

$$\psi(x) = \sum_{n \leq x} \Lambda(n) = \sum_{m=1}^{\infty} \sum_{\substack{p \\ p^m \leq x}} \Lambda(p^m) = \sum_{m=1}^{\infty} \sum_{p \leq x^{1/m}} \log p .$$

Por ser p primo, debe ser $p \geq 2$, luego la suma en p será vacía si $x^{1/m} < 2 \Leftrightarrow (1/m) \log x < \log 2 \Leftrightarrow m > \log x / \log 2 = \log_2 x$; es decir, la suma en m es finita, obteniendo

$$\psi(x) = \sum_{m \leq \log_2 x} \sum_{p \leq x^{1/m}} \log p .$$

Esto puede reescribirse en términos de la función definida a continuación.

Definición 2. Para todo entero $n \geq 1$ se define

$$\Lambda_1(n) = \begin{cases} \log p & , \text{ si } n \text{ es un primo } p \\ 0 & , \text{ en otro caso} \end{cases}$$

Además, para $x > 0$ se define la función ϑ de Chebyshev dada por

$$\vartheta(x) = \sum_{n \leq x} \Lambda_1(n) .$$

De este modo, se observa que puede expresarse

$$\vartheta(x) = \sum_{p \leq x} \log p , \quad \psi(x) = \sum_{m \leq \log_2 x} \vartheta(x^{1/m}) .$$

El siguiente teorema muestra la relación entre los cocientes $\psi(x)/x$ y $\vartheta(x)/x$, demostrando que si uno converge el otro también deberá hacerlo y además al mismo límite.

Proposición 1. Para $x > 0$, se cumple

$$0 \leq \frac{\psi(x)}{x} - \frac{\vartheta(x)}{x} \leq \frac{\log^2 x}{2\sqrt{x} \log 2} .$$

Además, se observa que esto implica

$$\lim_{x \to \infty} \left(\frac{\psi(x)}{x} - \frac{\vartheta(x)}{x} \right) = 0 .$$

Demostración. Para todo primo p, $\log p > 0$, de lo que se deduce que $\vartheta(x) \geq 0$. Al ser

$$\psi(x) - \vartheta(x) = \sum_{m \leq \log_2 x} \vartheta(x^{1/m}) - \vartheta(x) = \sum_{2 \leq m \leq \log_2 x} \vartheta(x^{1/m}) \ ,$$

entonces, en particular $0 \leq \psi(x) - \vartheta(x)$, luego para $x > 0$ se satisface $0 \leq \psi(x)/x - \vartheta(x)/x$. Se observa que $\vartheta(x)$ puede acotarse por $x \log x$, ya que

$$\vartheta(x) = \sum_{p \leq x} \log p \leq \sum_{p \leq x} \log x \leq x \log x \ .$$

De este modo

$$\psi(x) - \vartheta(x) = \sum_{2 \leq m \leq \log_2 x} \vartheta(x^{1/m}) \leq \sum_{2 \leq m \leq \log_2 x} x^{1/m} \log x^{1/m} \leq$$

Y se observa que para $x > 0$, $x^{1/m} \log x^{1/m}$ es no creciente en m, por lo que puede tomarse

$$\leq \sum_{2 \leq m \leq \log_2 x} x^{1/2} \log x^{1/2} \leq \log_2 x \ \sqrt{x} \ \log x^{1/2} = \frac{\sqrt{x}(\log x)^2}{2 \log 2} \ .$$

Dividiendo por x se obtiene el resultado. Ademas, como $\sqrt{x} \gg (\log x)^2$, al tomar límites se comprueba la observación.

$\square$

Teorema 2 (Identidad de Abel)**.** Sea $a(n)$ una función aritmética; es decir, una función real o compleja definida sobre $\mathbb{N}$ y sea $A(x) = \sum_{n \leq x} a(n)$, donde $A(x) = 0$ para $x < 1$. Si $f \in C^{1)}([y,x])$ con $0 < y < x$, entonces

$$\sum_{y < n \leq x} a(n) f(n) = A(x) f(x) - A(y) f(y) - \int_y^x A(t) f'(t) dt \ .$$

Además, se observa que si $y < 1$, resulta

$$\sum_{n \leq x} a(n) f(n) = A(x) f(x) - \int_1^x A(t) f'(t) dt \ .$$

Demostración. Sean $k = \lfloor x \rfloor$ y $m = \lfloor y \rfloor$ de modo que $A(x) = A(k)$ y $A(y) = A(m)$. De este modo,

$$\begin{aligned}
\sum_{y < n \leq x} a(n) f(n) &= \sum_{n=m+1}^{k} a(n) f(n) = \sum_{n=m+1}^{k} (A(n) - A(n-1)) f(n) \\
&= \sum_{n=m+1}^{k} A(n) f(n) - \sum_{n=m}^{k-1} A(n) f(n+1) \\
&= \sum_{n=m+1}^{k-1} A(n)(f(n) - f(n+1)) + A(k) f(k) - A(m) f(m+1) \\
&= -\sum_{n=m+1}^{k-1} A(n) \int_n^{n+1} f'(t) dt + A(k) f(k) - A(m) f(m+1) \\
&= -\int_{m+1}^{k} A(t) f'(t) dt + A(x) f(x) - \int_k^x A(t) f'(t) dt - A(y) f(y) - \int_y^{m+1} A(t) f'(t) dt \\
&= A(x) f(x) - A(y) f(y) - \int_y^x A(t) f'(t) dt \ .
\end{aligned}$$

Para comprobar la observación, basta ver que si $y < 1$, entonces $A(y) = 0$ por hipótesis. $\square$

Proposición 3. Para $x \geq 2$, se tiene

$$\vartheta(x) = \pi(x)\log x - \int_2^x \frac{\pi(t)}{t}dt \ , \quad \pi(x) = \frac{\vartheta(x)}{\log x} + \int_2^x \frac{\vartheta(t)}{t\log^2 t}dt \ .$$

Demostración. Siendo $a(n)$ la función característica de los números primos; es decir,

$$a(n) = \begin{cases} 1 & , \text{ si } n \text{ es primo} \\ 0 & , \text{ en otro caso} \end{cases}$$

entonces pueden expresarse $\pi(x)$ y $\vartheta(x)$ como:

$$\pi(x) = \sum_{1<n\leq x} a(n) \ , \quad \vartheta(x) = \sum_{1<n\leq x} a(n)\log n \ .$$

Puede aplicarse la Identidad de Abel para obtener la expresión de $\vartheta(x)$, eligiendo $f(x) = \log x$, $y = 1$, y notando que $A(x) = \pi(x)$, obteniendo:

$$\vartheta(x) = \sum_{1<n\leq x} a(n)\log n = \pi(x)\log x - \pi(1)\log 1 - \int_1^x \pi(t)(\log t)' dt = \pi(x)\log x - \int_2^x \frac{\pi(t)}{t}dt \ ,$$

donde la última igualdad se da por ser $\pi(t) = 0$ para $t < 2$.

Por otro lado, eligiendo $b(n) = a(n)\log(n)$, $\pi(x)$ y $\vartheta(x)$ pueden expresarse como:
(se empieza desde $3/2$, que satisface $1 < 3/2 < 2$ para garantizar que el logaritmo esté bien definido)

$$\pi(x) = \sum_{3/2<n\leq x} \frac{b(n)}{\log n} \ , \quad \vartheta(x) = \sum_{1<n\leq x} b(n) \ .$$

Aplicando la Identidad de Abel para obtener la expresión de $\pi(x)$ eligiendo $f(x) = 1/\log x$, $y = 3/2$, y notando que $A(x) = \vartheta(x)$, se obtiene:

$$\pi(x) = \sum_{1<n\leq x} b(n)\log n = \frac{\vartheta(x)}{\log x} - \frac{\vartheta(3/2)}{\log 3/2} - \int_{3/2}^x \pi(t)(1/\log t)' dt = \frac{\vartheta(x)}{\log x} + \int_2^x \frac{\vartheta(t)}{t\log^2 t}dt \ ,$$

donde la última igualdad se da por ser $\vartheta(t) = 0$ para $t < 2$. □

Teorema 4. Las siguientes relaciones son equivalentes:

$$\lim_{x\to\infty} \frac{\pi(x)\log x}{x} = 1 \ , \qquad \lim_{x\to\infty} \frac{\vartheta(x)}{x} = 1 \ , \qquad \lim_{x\to\infty} \frac{\psi(x)}{x} = 1 \ .$$

Demostración. Se denotan por $a)$, $b)$, $c)$ las respectivas relaciones del enunciado.

Por la Proposición 3

$$\frac{\vartheta(x)}{x} = \frac{\pi(x)\log x}{x} - \frac{1}{x}\int_2^x \frac{\pi(t)}{t}dt \ , \quad \frac{\pi(x)\log x}{x} = \frac{\vartheta(x)}{x} + \frac{\log x}{x}\int_2^x \frac{\vartheta(t)}{t\log^2 t}dt \ .$$

De este modo, para ver que $a) \Rightarrow b)$ basta mostrar que $a)$ implica

$$\lim_{x\to\infty} \frac{1}{x}\int_2^x \frac{\pi(t)}{t}dt = 0 \ ,$$

pero puesto que por $a)$, $\pi(t)/t = O(1/\log t)$ para $t \geq 2$, entonces

$$\frac{1}{x}\int_2^x \frac{\pi(t)}{t}dt = O\left(\frac{1}{x}\int_2^x \frac{dt}{\log t}\right) \ .$$

Ahora bien,

$$\frac{1}{x}\int_2^x \frac{dt}{\log t} = \frac{1}{x}\int_2^{\sqrt{x}} \frac{dt}{\log t} + \frac{1}{x}\int_{\sqrt{x}}^x \frac{dt}{\log t} \leq \frac{1}{x}\frac{\sqrt{x}}{\log 2} + \frac{1}{x}\frac{x-\sqrt{x}}{\log\sqrt{x}} \longrightarrow 0\ , \quad \text{cuando } x \to \infty\ .$$

Por otro lado, para ver $b) \Rightarrow a)$, basta mostrar que $b)$ implica

$$\lim_{x\to\infty} \frac{\log x}{x}\int_2^x \frac{\vartheta(t)}{t\log^2 t}dt = 0\ ,$$

pero puesto que por $b)$, $\vartheta(t) = O(t)$, entonces

$$\frac{\log x}{x}\int_2^x \frac{\vartheta(t)}{t\log^2 t}dt = O\left(\frac{\log x}{x}\int_2^x \frac{dt}{\log^2 t}\right)\ .$$

Ahora bien, de forma análoga

$$\frac{1}{x}\int_2^x \frac{dt}{\log^2 t} = \frac{1}{x}\int_2^{\sqrt{x}} \frac{dt}{\log^2 t} + \frac{1}{x}\int_{\sqrt{x}}^x \frac{dt}{\log^2 t} \leq \frac{1}{x}\frac{\sqrt{x}}{\log^2 2} + \frac{1}{x}\frac{x-\sqrt{x}}{\log^2\sqrt{x}} \longrightarrow 0\ , \quad \text{cuando } x \to \infty\ .$$

Por último, $b)$ y $c)$ son equivalentes por la Proposición 1, por lo que se demuestra el Teorema. □

1.2. Expresión asintótica de las sumas parciales de la serie de los recíprocos de los primos

Euler [5] (1737) demostró que la serie de los recíprocos de los primos $\sum 1/p$ diverge, hallando además que este hecho implica el Teorema de Euclides sobre la existencia de infinitos primos. En esta sección se obtendrá una fórmula asintótica para las sumas parciales de esta serie.

Observación. Resulta útil tener presente la siguiente relación:

$$\sum_{n\leq x} \log n = \log 1 + \log 2 + \cdots + \log\lfloor x\rfloor = \log(1\cdot 2\cdots\lfloor x\rfloor) = \log\lfloor x\rfloor!\ .$$

Proposición 5. Para $x \geq 1$, se tiene

$$\sum_{n\leq x} \Lambda(n)\left\lfloor\frac{x}{n}\right\rfloor = \log\lfloor x\rfloor!\ .$$

Demostración. Primero se observa que $\sum_{d|n}\Lambda(d) = \log n$. Esto se debe a que, por el Teorema Fundamental de la Aritmética, todo $n \geq 1$ puede expresarse como $n = p_1^{k_1}\cdots p_s^{k_s}$, y si $d|n$, entonces $d = p_1^{m_1}\cdots p_s^{m_s}$ con $m_j \leq k_j \quad \forall j$. De este modo, como $\Lambda(d) \neq 0$ solo cuando d es una potencia de un primo, resulta

$$\sum_{d|n}\Lambda(d) = \sum_{j=1}^{s} k_j \log p_j = \sum_{j=1}^{s}\log p_j^{k_j} = \log\left(\prod_{j=1}^{s} p_j^{k_j}\right) = \log n\ .$$

Como $\lfloor x/n\rfloor$ es la cantidad de enteros $\leq x$ que son divisibles por n; es decir, $\lfloor x/n\rfloor = \sum_{d\leq x;\ d|n} 1$. Entonces,

$$\sum_{n\leq x}\Lambda(n)\left\lfloor\frac{x}{n}\right\rfloor = \sum_{n\leq x}\Lambda(n)\sum_{\substack{d\leq x\\ d|n}} 1 = \sum_{n\leq x}\sum_{d|n}\Lambda(d) = \sum_{n\leq x}\log n\ .$$

Y el resultado se deduce de la Observación anterior. □

Proposición 6. Para todo $x \geq 2$

$$\log\lfloor x \rfloor ! = x\log x - x + O(\log x) \ .$$

Equivalentemente,

$$\sum_{n \leq x} \Lambda(n) \left\lfloor \frac{x}{n} \right\rfloor = x\log x - x + O(\log x) \ .$$

Demostración. Eligiendo $a(n) \equiv 1$, entonces $A(x) = \sum_{n \leq x} a(n) = \lfloor x \rfloor$. Tomando $f(t) = \log t$, $y = 1$ puede aplicarse la Identidad de Abel, obteniendo

$$\sum_{n \leq x} \log n = \lfloor x \rfloor \log x - \lfloor 1 \rfloor \log 1 - \int_1^x \lfloor t \rfloor (\log t)' dt = \lfloor x \rfloor \log x - \int_1^x \frac{\lfloor t \rfloor}{t} dt = \lfloor x \rfloor \log x - \int_1^x dt + \int_1^x \frac{t - \lfloor t \rfloor}{t} dt \ .$$

Y puesto que

$$\int_1^x \frac{t - \lfloor t \rfloor}{t} dt = O\left(\int_1^x \frac{1}{t} dt \right) = O(\log x) \ ,$$

entonces

$$\sum_{n \leq x} \log n = x\log x - x + 1 + O(\log x) = x\log x - x + O(\log x) \ .$$

Por la Observación anterior y la Proposición 5, se tiene el resultado. □

Observación. Es interesante notar que en particular, la Proposición 6 implica que $\forall x \geq 2$

$$\log\lfloor x \rfloor ! = x\log x + O(x) \ .$$

Proposición 7. Para todo $x \geq 2$

$$\sum_{p \leq x} \log p \left\lfloor \frac{x}{p} \right\rfloor = x\log x + O(x) \ .$$

Equivalentemente,

$$\sum_{n \leq x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor = x\log x + O(x) \ .$$

Demostración. Puesto que $\Lambda(n) = 0$ salvo cuando $n = p^m$ con p primo y $m \geq 1$, entonces

$$\sum_{n \leq x} \Lambda(n) \left\lfloor \frac{x}{n} \right\rfloor = \sum_{m=1}^{\infty} \sum_{\substack{p \\ p^m \leq x}} \Lambda(p^m) \left\lfloor \frac{x}{p^m} \right\rfloor \ .$$

Dado que $p^m \leq x \Rightarrow p \leq x$, y $\lfloor x/p^m \rfloor = 0$ si $p > x$, se tiene

$$\sum_{n \leq x} \Lambda(n) \left\lfloor \frac{x}{n} \right\rfloor = \sum_{p \leq x} \sum_{m=1}^{\infty} \log p \left\lfloor \frac{x}{p^m} \right\rfloor = \sum_{p \leq x} \log p \left\lfloor \frac{x}{p} \right\rfloor + \sum_{p \leq x} \sum_{m=2}^{\infty} \log p \left\lfloor \frac{x}{p^m} \right\rfloor \ .$$

Puesto que el segundo sumando cumple

$$\sum_{p \leq x} \sum_{m=2}^{\infty} \log p \left\lfloor \frac{x}{p^m} \right\rfloor = \sum_{p \leq x} \log p \sum_{m=2}^{\infty} \left\lfloor \frac{x}{p^m} \right\rfloor \leq \sum_{p \leq x} \log p \sum_{m=2}^{\infty} \frac{x}{p^m} = x \sum_{p \leq x} \log p \sum_{m=2}^{\infty} (1/p)^m =$$

Al ser $\sum_{m \geq 2} w^m = w^2/(1-w)$ para todo w tal que $|w| < 1$,

$$= x \sum_{p \leq x} \log p \frac{1}{p^2} \frac{1}{1 - 1/p} = x \sum_{p \leq x} \frac{\log p}{p(p-1)} \leq x \sum_{n=2}^{\infty} \frac{\log n}{n(n-1)} \ .$$

Dado que $\sum_{n=2}^{\infty} \frac{\log n}{n(n-1)} \le \sum_{n=2}^{\infty} \frac{n^{1/2}}{n^2} < \infty$, entonces es claro que

$$\sum_{n\le x} \Lambda(n) \left\lfloor \frac{x}{n} \right\rfloor = \sum_{p\le x} \log p \left\lfloor \frac{x}{p} \right\rfloor + O(x) \ ,$$

y por la Proposición 6 queda demostrado el resultado.

□

Lema 8 (Lemas técnicos sobre la función parte entera por defecto)**.**

i) $\lfloor 2x \rfloor - 2\lfloor x \rfloor \in \{0,1\}, \quad \forall x \in \mathbb{R}$

ii) $\lfloor x+n \rfloor = \lfloor x \rfloor + n \ , \quad \forall x \in \mathbb{R}, n \in \mathbb{Z}$

iii) $\left\lfloor \frac{\lfloor x/n \rfloor}{m} \right\rfloor = \left\lfloor \frac{x}{nm} \right\rfloor \ , \quad \forall x \in \mathbb{R}, n,m \in \mathbb{N}$

Demostración.

i) Para cierto $k \in \mathbb{Z}$, debe ser $x \in [k, k+1/2)$ o bien $x \in [k+1/2, k+1)$.
En el primer caso, $\lfloor 2x \rfloor = 2k$ y $2\lfloor x \rfloor = 2k$, luego $\lfloor 2x \rfloor - 2\lfloor x \rfloor = 0$.
En el segundo, $\lfloor 2x \rfloor = 2k+1$ y $2\lfloor x \rfloor = 2k$, luego $\lfloor 2x \rfloor - 2\lfloor x \rfloor = 1$.

ii) Por definición $\lfloor x+n \rfloor \le x+n < \lfloor x+n \rfloor + 1$, luego $\lfloor x+n \rfloor - n \le x < \lfloor x+n \rfloor - n + 1$. Esta última igualdad implica $\lfloor x \rfloor = \lfloor x+n \rfloor - n$, por lo que $\lfloor x \rfloor + n = \lfloor x+n \rfloor$.

iii) Si $\lfloor x/n \rfloor = x/n$ es inmediato. En caso contrario, debe ser $\lfloor x/n \rfloor < x/n$, luego $\lfloor x/n \rfloor / m < x/(nm)$. En particular, se satisfará $\lfloor \lfloor x/n \rfloor / m \rfloor \le \lfloor x/(nm) \rfloor$. Si se cumple la desigualdad estricta $\lfloor \lfloor x/n \rfloor / m \rfloor < \lfloor x/(nm) \rfloor$ debe existir un valor k tal que $\lfloor x/n \rfloor < k \le x/n$ con $k/m = \lfloor x/(nm) \rfloor$; es decir, $k = m\lfloor x/(nm) \rfloor$por lo que es entero. Como tal entero satisface $\lfloor x/n \rfloor < k \le x/n$, necesariamente debe ser $k = x/n$, pero en ese caso x/n es entero y se cumple $\lfloor x/n \rfloor = x/n$, llegando a una contradicción.

□

Teorema 9 (Tauberiano de Shapiro)**.** Para todo $x \ge 1$ se cumple

$$\sum_{p\le x} \frac{\log p}{p} = \log x + O(1) \ .$$

Además, existen constantes $A, B > 0$ y $x_0 > 0$ tales que

- $\vartheta(x) \le Bx, \quad \forall x \ge 1$
- $\vartheta(x) \ge Ax, \quad \forall x \ge x_0$

Demostración. Se demostrará primero el primer punto. Para ello, se observa que

$$\sum_{n\le x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor - 2 \sum_{n\le x/2} \Lambda_1(n) \left\lfloor \frac{x/2}{n} \right\rfloor = \sum_{n\le x/2} \Lambda_1(n) \left(\left\lfloor \frac{x}{n} \right\rfloor - 2\left\lfloor \frac{x}{2n} \right\rfloor \right) + \sum_{x/2<n\le x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor \ .$$

Por el Lema 8.i), se deduce que $\left\lfloor \frac{x}{n} \right\rfloor - 2\left\lfloor \frac{x}{2n} \right\rfloor$ vale 0 o 1, y por ser $\vartheta(x) \ge 0$ resulta

$$\sum_{n\le x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor \ge \sum_{x/2<n\le x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor \ge \sum_{x/2<n\le x} \Lambda_1(n) \ge \vartheta(x) - \vartheta(x/2) \ .$$

Ahora bien, por la Proposición 7 se cumple

$$\sum_{n\le x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor - 2 \sum_{n\le x/2} \Lambda_1(n) \left\lfloor \frac{x/2}{n} \right\rfloor = x\log x + O(x) - 2\left(\frac{x}{2} \log(x/2) + O(x/2) \right) = O(x) \ .$$

De este modo $O(x) \geq \vartheta(x) - \vartheta(x/2)$, luego existe $C > 0$ constante tal que $\vartheta(x) - \vartheta(x/2) \leq Cx$ para todo $x \geq 1$. Reemplazando sucesivamente x por $x/2, x/4, \ldots$, se obtiene

$$\vartheta(x/2) - \vartheta(x/4) \leq Cx/2 , \quad \vartheta(x/4) - \vartheta(x/8) \leq Cx/4 , \quad \ldots$$

Como para $2^n > x$ se tiene $\vartheta(x/2^n) = 0$, se observa que sumando las desigualdades anteriores se obtiene en el lado izquierdo una serie telescópica, donde a partir de cierto n todos los sumandos serán 0, cancelándose todos menos el primero; esto es

$$\vartheta(x) = \sum_{n \geq 1} \vartheta(x/2^n) - \vartheta(x/2^{n+1}) \leq Cx(1 + 1/2 + 1/4 + \ldots) = 2Cx .$$

Eligiendo $B = 2C$ se obtiene el resultado.

Para demostrar la primera parte, se expresa $\lfloor x/n \rfloor = x/n + O(1)$, y de este modo

$$\sum_{n \leq x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor = \sum_{n \leq x} \Lambda_1(n) \left(\frac{x}{n} + O(1) \right) = x \sum_{n \leq x} \frac{\Lambda_1(n)}{n} + O(\vartheta(x)) = x \sum_{n \leq x} \frac{\Lambda_1(n)}{n} + O(x) \ ,$$

donde $O(\vartheta(x)) = O(x)$ se deduce del punto anterior. Además, por la Proposición 7 puede expresarse

$$\sum_{n \leq x} \frac{\Lambda_1}{n} = \frac{1}{x} \left(\sum_{n \leq x} \Lambda_1(n) \left\lfloor \frac{x}{n} \right\rfloor + O(x) \right) = \frac{1}{x} (x \log x + O(x)) = \log x + O(1) \ .$$

Para demostrar el segundo punto, se observa que por la primera parte, puede escribirse $\sum_{n \leq x} \Lambda_1(n)/n = \log x + R(x)$ con $R(x) = O(1)$. Ahora bien, es claro que existe $M > 0$ con $|R(x)| \leq M$. Eligiendo $A \in (0,1)$,

$$\sum_{n \leq x} \frac{\Lambda_1(n)}{n} - \sum_{n \leq Ax} \frac{\Lambda_1(n)}{n} = \log x + R(x) - \log(Ax) - R(Ax) = -\log A + R(x) - R(Ax) \geq -\log A - 2M \ .$$

Como se satisface

$$\sum_{n \leq x} \frac{\Lambda_1(n)}{n} - \sum_{n \leq Ax} \frac{\Lambda_1(n)}{n} = \sum_{Ax < n \leq x} \frac{\Lambda_1(n)}{n} \leq \frac{1}{Ax} \sum_{n \leq x} \Lambda_1(n) = \frac{\vartheta(x)}{Ax} \ ,$$

entonces, eligiendo A tal que $-\log A - 2M = 1$; es decir, $A = e^{-2M-1}$ que pertenece a $(0,1)$, se observa

$$\vartheta(x) \geq Ax \ , \ \text{ cuando } x \geq x_0 := 1/A \ .$$

$\square$

Teorema 10. Existe una constante A tal que

$$\sum_{p \leq x} \frac{1}{p} = \log \log x + A + O\left(\frac{1}{\log x} \right), \quad \forall x \geq 2 \ .$$

Demostración. Sea

$$A(x) = \sum_{p \leq x} \frac{\log p}{p}$$

y sea $a(n)$ la función característica de los números primos; es decir,

$$a(n) = \begin{cases} 1, & \text{si } n \text{ es primo} \\ 0, & \text{en otro caso} \end{cases}$$

Entonces,

$$\sum_{p \leq x} \frac{1}{p} = \sum_{n \leq x} \frac{a(n)}{n} \ , \quad A(x) = \sum_{n \leq x} \frac{a(n)}{n} \log n \ .$$

De este modo, aplicando la Identidad de Abel con $f(t) = 1/\log t$, al ser $A(t) = 0$ para $t < 2$, se tiene

$$\sum_{p \le x} \frac{1}{p} = \frac{A(x)}{\log x} + \int_2^x \frac{A(t)}{t \log^2 t} dt \ .$$

Por el Teorema 9 puede ponerse $A(x) = \log(x) + R(x)$ con $R(x) = O(1)$, luego

$$\sum_{p \le x} \frac{1}{p} = \frac{\log x + O(1)}{\log x} + \int_2^x \frac{\log t + R(t)}{t \log^2 t} dt = 1 + O\left(\frac{1}{\log x}\right) + \int_2^x \frac{dt}{t \log t} + \int_2^x \frac{R(t)}{t \log^2 t} dt \ .$$

Ahora bien,

$$\int_2^x \frac{dt}{t \log t} = \int_2^x \frac{1/t}{\log t} dt = \log \log x - \log \log 2 \ ,$$

y puede expresarse

$$\int_2^x \frac{R(t)}{t \log^2 t} dt = \int_2^\infty \frac{R(t)}{t \log^2 t} dt - \int_x^\infty \frac{R(t)}{t \log^2 t} dt \ ,$$

donde resulta

$$\int_x^\infty \frac{R(t)}{t \log^2 t} dt = O\left(\int_x^\infty \frac{dt}{t \log^2 t}\right) = O\left(\frac{1}{\log x}\right) \ .$$

De este modo, se obtiene

$$\sum_{p \le x} \frac{1}{p} = \log \log x + A + O\left(\frac{1}{\log x}\right) \ , \ \ \text{donde } A := 1 - \log \log 2 + \int_2^\infty \frac{R(t)}{t \log^2 t} dt \ .$$

$\square$

Capítulo 2

Método de Mapes

Existe una amplia variedad de métodos que permiten obtener el valor de $\pi(x)$, tanto por aproximaciones como de forma exacta. Nuestro objetivo en este trabajo es obtener el valor exacto de $\pi(x)$ mediante el método de Mapes [12]. Este método y los que se describirán en el proceso permiten calcular $\pi(x)$ conociendo cierta cantidad de primos menores que x.

El método más antiguo de este tipo se debe a Eratóstenes (276-197 A.C.), director de la Biblioteca de Alejandría. Dicho método, llamado criba de Eratóstenes, ha sido durante mucho tiempo la forma más práctica de calcular $\pi(x)$, puesto que permite hallar todos los números primos menores o iguales a x, por lo que luego puede obtenerse $\pi(x)$ simplemente contándolos.

En la historia reciente destaca la fórmula de Legendre [10] (1830), la cual permite calcular $\pi(x)$ conociendo los primos hasta $\sqrt{x}$. El mismo Legendre calculó, con cierto error, el valor $\pi\left(10^6\right) = 78.498$.

Esta fórmula abrió muchas puertas en el cálculo del valor $\pi(x)$, puesto que surgieron métodos a partir de variaciones de la fórmula y reordenaciones inteligentes de sus términos. Por ejemplo, Meissel [13] (1870) calculó exitosamente con su método $\pi\left(10^7\right)$ y $\pi\left(10^8\right)$, y en 1885 publicó el valor $\pi\left(10^9\right)$, obteniendo 50.847.478. Lehmer [11] (1959), con la llegada de las computadoras y mejorando aún más el método, corrigió el valor de $\pi\left(10^9\right) = 50.847.534$. Además, computó con un IBM 701 el valor $\pi\left(10^{10}\right)$, obteniendo 455.052.512, excediéndose solo por 1 del valor real.

Aún con la potencia de las computadoras, el coste en tiempo de todos estos métodos es realmente elevado por lo que las contribuciones de Lehmer y Mapes [12] (1963) para su optimización resultan fundamentales.

Recientemente, destacan las publicaciones de Bohman [2] (1972), Lagarias, Miller, Odlyzko [8] (1985) y Deléglise, Rivat [4] (1996).

2.1. La criba de Eratóstenes

En este capítulo, se tomará $x \in \mathbb{N}$ a la hora de calcular $\pi(x)$, aunque cabe destacar que esto no influye a ningún desarrollo, pues bastaría tomar $\pi(\lfloor x \rfloor)$ en el caso $x \in \mathbb{R}^+$.

Eratóstenes formuló un método, llamado criba de Eratóstenes, que permite hallar todos los números primos menores o iguales que x a base de eliminar los números compuestos en el intervalo $[1,x]$. Esto es, para cada primo $p \in [1,x]$ se deben cribar sus múltiplos $kp \leq x$ con $k > 1$. La proposición siguiente permite reducir la cantidad números primos que necesitamos explorar y también sus múltiplos a eliminar.

Proposición 11. Cualquier número compuesto x tiene al menos un factor primo menor o igual que $\sqrt{x}$.

Demostración. Sea p el menor factor primo de x y supongamos que $p > \sqrt{x}$. Como $p|x$, $\exists q$ tal que $x = pq$, y al ser p el menor factor primo,

$$x = pq > p^2 > (\sqrt{x})^2 = x .$$

$\square$

Observación. A la hora de buscar números compuestos en [1,x], de la Proposición 11 se deduce:

- Para cada primo p basta buscar sus múltiplos de la forma $p^2 + kp$ para $k \geq 0$.
- Además, basta hacerlo solo para los primos p con $p < \sqrt{x}$.

Teniendo en cuenta lo anterior, el algoritmo de la criba de Eratóstenes para obtener los números primos menores o iguales a x es el siguiente:

1. Se crea una tabla en la que aparezcan todos los valores en $[2,x]$ ordenados de forma ascendente.
2. Se selecciona el menor valor p de la tabla no marcado como primo y se marca. Si $p \geq \sqrt{x}$, se pasa al paso siguiente. En caso contrario, se eliminan secuencialmente los elementos $p^2 + kp$ con $k \geq 0$ hasta alcanzar el final de la tabla y se repite este paso.
3. Se finaliza el algoritmo. Los elementos marcados como primos y los no eliminados de la tabla son los números primos menores o iguales que x.

Ejemplo. Para $x = 20$ los pasos del algoritmo de la criba de Eratóstenes son los siguientes:

Primero se construye la tabla y se observa que $\sqrt{x} = \sqrt{20} \approx 4,47$.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Se selecciona y se marca el menor valor $p = 2$, se comprueba $p < \sqrt{x}$ y se eliminan sus múltiplos desde $p^2 = 4$. Estos son: 4,6,8,10,12,14,16,18,20.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Se selecciona y se marca el menor valor $p = 3$, se comprueba $p < \sqrt{x}$ y se eliminan sus múltiplos desde $p^2 = 9$. Estos son: 6,9,12,15,18.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Se selecciona y se marca el menor valor $p = 5$, se observa que $p \geq \sqrt{x}$, por lo que se finaliza el algoritmo. Todos los valores marcados y los no eliminados son primos.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

Así, los números primos menores o iguales a 20 son: 2,3,5,7,11,13,17,19. De este modo $\pi(20) = 8$.

Nota. Las ideas anteriores sugieren que para calcular los primos menores o iguales que x es necesario almacenar en memoria una tabla de tamaño x. No obstante, podemos cribar en primer lugar, con el método descrito, el subintervalo $[2, \sqrt{x}]$ y, con los primos obtenidos, realizar una criba parcial de $(\sqrt{x}, x]$ utilizando subintervalos de longitud $\sqrt{x}$. Con esta modificación se puede probar que el método de la criba de Eratóstenes tiene un coste en tiempo de orden $O(x \log \log x)$ y en espacio de $O(\sqrt{x})$.

El método de Mapes requiere conocer cierta cantidad de primos. Por este motivo, nuestra implementación incluye, en la clase *Mapes*, las funciones *eratosthenes* y *eratosthenesInterval* que realizan el cribado total y parcial de un intervalo (ver Apéndice B.3).

2.2. Fórmula de Legendre

Legendre halló una fórmula que explota la Proposición 11 para calcular el valor de $\pi(x)$. Para dar la fórmula es necesario introducir la suma de Legendre.

Notación. En adelante se denotará el n-ésimo primo por p_n. Además, se observa que $\pi(p_n) = n$ y para x con $p_n \leq x < p_{n+1}$, $\pi(x) = n$.

Definición 3 (Suma de Legendre)**.** Para cada $a \geq 1$ se define la suma de Legendre como la aplicación

$$\begin{aligned} \phi :\ & \mathbb{N} \longrightarrow \mathbb{N} \\ & x \longmapsto \phi(x,a) := \#\{n \leq x;\ p|n \Rightarrow p > p_a\} \end{aligned}$$

Se observa que $\phi(x,a)$ cuenta los enteros positivos $\leq x$ que no son divisibles por ninguno de los a primeros primos $p_1, p_2, \ldots, p_a$; en particular $\phi(x,a) \geq 1$ para todo $x, a > 0$, ya que este recuento incluye al entero 1. La fórmula de Legendre aprovecha este hecho para expresar el valor $\pi(x)$ en función de la suma de Legendre.

Proposición 12 (Fórmula de Legendre)**.** Sea $a = \pi(\sqrt{x})$, entonces

$$\pi(x) = a - 1 + \phi(x,a)\ .$$

Demostración. Por la Proposición 11 se sabe que los números compuestos en $[1,x]$ tienen como factores los primeros a primos, por lo que se deduce que el número de primos en $[1,x]$ es exactamente $\pi(x) = a + \phi(x,a) - 1$. □

Mediante el Principio de Inclusión-Exclusión, que se recuerda en el Lema siguiente, puede expresarse la función $\phi(x,a)$ a partir de sumas, hecho que le otorga el nombre.

Lema 13. (Principio de Inclusión-Exclusión) Si $A_1, \ldots, A_n$ son conjuntos finitos, entonces:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{1\leq i\leq n} |A_i| - \sum_{1\leq i<j\leq n} |A_i \cap A_j| + \sum_{1\leq i<j<k\leq n} |A_i \cap A_j \cap A_k| + \cdots + (-1)^{n+1}|A_1 \cap \cdots \cap A_n|\ .$$

Proposición 14.

$$\phi(x,a) = x - \sum_{1\leq i\leq a} \left\lfloor \frac{x}{p_i} \right\rfloor + \sum_{1\leq i<j\leq a} \left\lfloor \frac{x}{p_i p_j} \right\rfloor - \sum_{1\leq i<j<k\leq a} \left\lfloor \frac{x}{p_i p_j p_k} \right\rfloor + \cdots + (-1)^a \left\lfloor \frac{x}{p_1 p_2 \cdots p_a} \right\rfloor\ .$$

Demostración. Primero, se observa que $\lfloor x/q \rfloor$ es la cantidad de enteros en $[1,x]$ divisibles por q. De este modo, en $[1,x]$ el número de enteros divisibles por el producto de:

- exactamente un primo p_i es $\left\lfloor \frac{x}{p_i} \right\rfloor$.
- exactamente dos primos p_i, p_j es $\left\lfloor \frac{x}{p_i p_j} \right\rfloor$.

  ...
- todos los a primeros primos es $\left\lfloor \frac{x}{p_1 p_2 \cdots p_a} \right\rfloor$.

Ahora bien, como los elementos en $\{n \leq x;\ p|n \Rightarrow p > p_a\}$ son aquellos en $[1,x]$ que no son divisibles por ningún producto de los primeros a primos, para obtener el valor $\phi(x,a)$ basta restar a x, que es el número total de elementos en $[1,x]$, el número de elementos que sí son divisibles por algún producto, valor que puede obtenerse mediante el Principio de Inclusión-Exclusión:

$$\begin{aligned} \phi(x,a) &= \#\{n \leq x;\ p|n \Rightarrow p > p_a\} = \\ &= x - \sum_{1\leq i\leq a} \left\lfloor \frac{x}{p_i} \right\rfloor + \sum_{1\leq i<j\leq a} \left\lfloor \frac{x}{p_i p_j} \right\rfloor - \sum_{1\leq i<j<k\leq a} \left\lfloor \frac{x}{p_i p_j p_k} \right\rfloor + \cdots + (-1)^a \left\lfloor \frac{x}{p_1 p_2 \cdots p_a} \right\rfloor\ . \end{aligned}$$

□

Observación. La Proposición anterior permite expresar la fórmula de Legendre del siguiente modo

$$\pi(x) = a - 1 + \phi(x,a) =$$

$$= \pi\left(\sqrt{x}\right) - 1 + x - \sum_{p_i \le \sqrt{x}} \left\lfloor \frac{x}{p_i} \right\rfloor + \sum_{p_i < p_j \le \sqrt{x}} \left\lfloor \frac{x}{p_i p_j} \right\rfloor - \cdots + (-1)^a \left\lfloor \frac{x}{p_1 p_2 \cdots p_a} \right\rfloor .$$

Nota. Puede demostrarse que el método de Legendre tiene un coste en tiempo de orden $O(x)$ y en espacio de $O(\sqrt{x})$.

2.3. Métodos de Meissel y Lehmer

Se observa que para poder calcular $\pi(x)$ mediante la fórmula de Legendre es necesario conocer de antemano los primeros $a = \pi(\sqrt{x})$ primos. Además, en $\phi(x,a)$ aparece un total de 2^{a-1} sumandos. Por este motivo, se definen los valores $P_k(x,a)$, que pretenden disminuir el coste de cálculo de $\pi(x)$ a base de reducir el valor de a.

Definición 4. $P_k(x,a)$ es el número de enteros en el intervalo $[1,x]$ que pueden expresarse como producto de k primos mayores que p_a (no necesariamente distintos). Esto es

$$P_k(x,a) = \#\{n = p_{i_1} p_{i_2} \cdots p_{i_k} \le x \ ; \ \ a < i_1 \le i_2 \le \cdots \le i_k\} \ .$$

Además, se establece $P_0(x,a) = 1$.

Proposición 15. Para $m \ge 1$, si a es tal que $x^{1/m} < p_{a+1}$, entonces $P_k(x,a) = 0 \ , \ \ \forall k \ge m$.
Para $m > 1$, si a es tal que $p_{a+1} \le x^{1/(m-1)}$, entonces $P_k(x,a) \ne 0 \ , \ \ \forall k < m$.

Demostración. Si $p_{a+1} > x^{1/m}$, entonces $p_{i_1} \cdots p_{i_k} > x^{k/m} \quad \forall i_1, \ldots, i_k > a$, y como $x^{k/m} > x$ para $k \ge m$, entonces $P_k(x,a) = 0$ para $k \ge m$.

Por otro lado, si $p_{a+1} \le x^{1/(m-1)}$, en particular $p_{a+1}^k \le x^{k/(m-1)} \le x$ para $k < m$, por lo que debe ser $P_k(x,a) \ne 0$ para $k < m$.

□

La siguiente propiedad muestra que los valores $P_k(x,a)$ resultan ser una reagrupación de los términos en $\phi(x,a)$.

Proposición 16.

$$\phi(x,a) = \sum_{k=0}^{\infty} P_k(x,a) \ .$$

Demostración. La igualdad se deduce de las definiciones de $\phi(x,a)$ y $P_k(x,a)$.

$\phi(x,a)$ es la cantidad de enteros positivos $\le x$ que no son divisibles por ninguno de los primeros a primos, mientras que $P_k(x,a)$ para $k \ge 1$ es la cantidad de enteros positivos $\le x$ que son divisibles por exactamente k primos mayores que p_a. De este modo, la suma $P_1(x,a) + P_2(x,a) + \ldots$ contará la cantidad de enteros en el intervalo $[1,x]$ que puede ponerse como producto de cualquier cantidad de primos mayores que p_a.

La única diferencia entre $\phi(x,a)$ y $P_1(x,a) + P_2(x,a) + \ldots$ es la siguiente: 1 no es divisible por ningún primo así que aparece contabilizado en $\phi(x,a)$, pero al no poder ponerse como producto de primos, no se ha tenido en cuenta en $P_1(x,a) + P_2(x,a) + \ldots$.

Ahora bien, como se ha definido $P_0(x,a) = 1$, se concluye que

$$\phi(x,a) = \sum_{k=0}^{\infty} P_k(x,a) \ .$$

□

Observación. Vale la pena notar que por la Proposición 15, la suma en la Proposición 16 es de una cantidad finita de términos $P_k(x,a)$ no nulos.

Proposición 17. Para $k \geq 2$ se cumple

$$P_k(x,a) = \sum_{i>a} P_{k-1}\left(\frac{x}{p_i}, i-1\right) .$$

Demostración. Para poder expresar $P_k(x,a)$ en función de $P_{k-1}(x,a)$, se dejarán variar los valores $i_2,\ldots,i_k$ y se fijarán los posibles valores $i_1 > a$; es decir,

$$\begin{aligned} P_k(x,a) &= \#\{n = p_{i_1}p_{i_2}\cdots p_{i_k} \leq x\ ;\ \ a < i_1 \leq i_2 \leq \cdots \leq i_k\} = \\ &= \sum_{i_1>a} \#\{n = p_{i_1}p_{i_2}\cdots p_{i_k} \leq x\ ;\ \ i_1 \leq i_2 \leq \cdots \leq i_k\} = \\ &= \sum_{i_1>a} \#\{n = p_{i_2}\cdots p_{i_k} \leq x/p_{i_1}\ ;\ \ i_1 - 1 < i_2 \leq \cdots \leq i_k\} = \sum_{i>a} P_{k-1}\left(x/p_i, i-1\right) . \end{aligned}$$

□

Gracias a esta propiedad, a partir del valor $P_1(x,a)$ pueden obtenerse el resto, aunque solo utilizaremos los calculados a continuación.

Proposición 18. Los valores de $P_k(x,a)$ para $k = 1,2,3$ son:

i) $P_1(x,a) = \pi(x) - a$.

ii) $P_2(x,a) = \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1)$, donde $b := \pi(\sqrt{x})$.

iii) $P_3(x,a) = \sum_{i=a+1}^{c}\sum_{j=i}^{b_i} \pi\left(\frac{x}{p_ip_j}\right) - (j-1)$, donde $a < c := \pi\left(x^{1/3}\right)$ y $b_i := \pi\left(\sqrt{x/p_i}\right)$.

Demostración.

i) Puesto que $P_1(x,a)$ representa el número de primos p mayores que p_a y menores o iguales a x, entonces se deduce que $P_1(x,a) = \pi(x) - a$.

ii) Para evaluar $P_2(x,a)$ se usa la Proposición 17 y el apartado anterior, con lo que resulta

$$P_2(x,a) = \sum_{i>a} P_1\left(\frac{x}{p_i}, i-1\right) = \sum_{i>a} \pi\left(\frac{x}{p_i}\right) - (i-1) .$$

Siendo $b = \pi(\sqrt{x})$, se observa que para todo $i > b$ se cumple $p_i > \sqrt{x} \Rightarrow p_i^2 > x \Rightarrow p_i > x/p_i$ y por la Proposición 15 se deduce que los términos $P_1(x/p_i, i-1)$ son todos nulos para $i > b$. De este modo,

$$P_2(x,a) = \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) .$$

iii) Para evaluar $P_3(x,a)$ se usa la Proposición 17, obteniendo

$$P_3(x,a) = \sum_{i>a} P_2\left(\frac{x}{p_i}, i-1\right) = \sum_{i>a}\sum_{j>i-1} P_1\left(\frac{x/p_i}{p_j}, j-1\right) = \sum_{i>a}\sum_{j\geq i} \pi\left(\frac{x}{p_ip_j}\right) - (j-1) .$$

De forma análoga al apartado anterior, siendo $b_i = \pi\left(\sqrt{x/p_i}\right)$, se observa que para todo $j > i$ se cumple $p_j > \sqrt{x/p_i} \Rightarrow p_j^2 > x/p_i \Rightarrow p_j > x/(p_ip_j)$ y por la Proposición 15 se deduce que los términos $P_1(x/(p_ip_j), j-1)$ son todos nulos para $j > i$. Además, siendo $c = \pi(x^{1/3})$, se observa que para todo $i > c$ se cumple $p_i > x^{1/3} \Rightarrow p_i^{3/2} > x^{1/2} \Rightarrow p_i > (x/p_i)^{1/2}$ y por la Proposición 15 se deduce que los términos $P_2(x/p_i, i-1)$ son todos nulos para $i > c$. De este modo,

$$P_3(x,a) = \sum_{i=a+1}^{c} \sum_{j=i}^{b_i} \pi\left(\frac{x/p_i}{p_j}\right) - (j-1) \; .$$

□

Los métodos de Meissel y Lehmer aprovechan las Proposiciones 15 y 18 anteriores para calcular $\pi(x)$ eligiendo valores de a apropiados. El objetivo es reducir el valor de a, pero, como contrapartida, por la Proposición 15 esto aumentará el número de sumandos $P_k(x,a)$ que deben calcularse.

Teorema 19. (Método de Meissel) Sea $a = \pi(x^{1/3})$, entonces

$$\begin{aligned} \pi(x) &= a - 1 + \phi(x,a) - P_2(x,a) = \\ &= a - 1 + \phi(x,a) - \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) \; . \end{aligned}$$

Demostración. Siendo $a = \pi(x^{1/3})$, en particular $a < b := \pi(\sqrt{x})$, luego por la Proposición 18 resulta:

$$P_2(x,a) = \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) \; .$$

Además, al ser $a = \pi(x^{1/3}) < b = \pi(\sqrt{x})$, por la Proposición 15,

$$P_k(x,a) \neq 0 \; , \;\; \forall k < 3 \qquad y \qquad P_k(x,a) = 0 \; , \;\; \forall k \geq 3 \; ,$$

luego de la Proposición 16 se deduce

$$\begin{aligned} \phi(x,a) &= P_0(x,a) + P_1(x,a) + P_2(x,a) = \\ &= 1 + \pi(x) - a + \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) \; . \end{aligned}$$

Reordenando los términos, se obtiene el resultado. □

Teorema 20. (Método de Lehmer) Sea $a = \pi(x^{1/4})$, entonces

$$\begin{aligned} \pi(x) &= \phi(x,a) + a - 1 - P_2(x,a) - P_3(x,a) = \\ &= \phi(x,a) + a - 1 - \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) - \sum_{i=a+1}^{c} \sum_{j=i}^{b_i} \pi\left(\frac{x}{p_ip_j}\right) - (j-1) \; . \end{aligned}$$

Demostración. Siendo $a = \pi(x^{1/4})$, en particular $a < c := \pi(x^{1/3})$. Tomando $b_i := \pi\left(\sqrt{x/p_i}\right)$, por la Proposición 18 resulta:

$$P_3(x,a) = \sum_{i=a+1}^{c} \sum_{j=i}^{b_i} \pi\left(\frac{x}{p_ip_j}\right) - (j-1) \; .$$

Además, ser $a = \pi(x^{1/4}) < c = \pi(x^{1/3})$, por la Proposición 15,

$$P_k(x,a) \neq 0 \; , \;\; \forall k < 4 \qquad y \qquad P_k(x,a) = 0 \; , \;\; \forall k \geq 4 \; ,$$

luego de la Proposición 16 se deduce

$$\begin{aligned} \phi(x,a) &= P_0(x,a) + P_1(x,a) + P_2(x,a) + P_3(x,a) = \\ &= 1 + \pi(x) - a + \sum_{i=a+1}^{b} \pi\left(\frac{x}{p_i}\right) - (i-1) + \sum_{i=a+1}^{c} \sum_{j=i}^{b_i} \pi\left(\frac{x}{p_ip_j}\right) - (j-1) \; . \end{aligned}$$

Reordenando los términos, se obtiene el resultado. □

Es interesante observar que el método de Meissel, con $a = \pi\big(x^{1/3}\big)$ y el de Lehmer, con $a = \pi\big(x^{1/4}\big)$ reducen considerablemente el valor de a con respecto al de método de Legendre.

Nota. En [16] se menciona el método de Meissel tiene un coste en tiempo de orden $O\big(x/(\ln x)^3\big)$ y en espacio de $O\big(x^{1/2}/\ln x\big)$, mientras que el método de Lehmer tiene en tiempo $O\big(x/(\ln x)^4\big)$ y en espacio $O\big(x^{1/3}/\ln x\big)$.

2.4. Cálculo de la suma de Legendre

Más adelante veremos que el método de Mapes utiliza ciertas tablas mediante las que es posible calcular el valor de $\phi(x,a)$ para valores arbitrarios de $x \in \mathbb{N}$ y a menor que una cierta constante. La obtención de estas tablas y su uso explotan algunas propiedades de la función $\phi(x,a)$ que introducimos a continuación. Antes es necesario introducir los siguientes conceptos.

Nota. La función $\phi(x,a)$ puede definirse en $a = 0$ del siguiente modo:

$$\phi(x,0) = x \ .$$

Además, el dominio de $\phi(x,a)$ puede extenderse a $\mathbb{Z}$ tomando:

$$\begin{aligned} &\phi(0,a) = 0 \ , \\ &\phi(x,a) = -\phi(-x,a), \quad \forall x < 0 \ . \end{aligned}$$

Definición 5. Se define m_a como

$$m_a = p_1 p_2 \cdots p_a \ .$$

Definición 6. Se define la función indicatriz de Euler como

$$\begin{aligned} \varphi : \mathbb{N} &\longrightarrow \mathbb{N} \\ n &\longmapsto \#\{m \in \mathbb{N};\ m \leq n \wedge \operatorname{mcd}(n,m) = 1\} \end{aligned}$$

Lema 21. La función indicatriz satisface:

i) $\varphi(nm) = \varphi(n)\varphi(m)$, $\forall n,m$ tal que $\operatorname{mcd}(n,m) = 1$.

ii) $\varphi(p) = p-1, \quad \forall p$ primo.

Corolario 22. La función indicatriz evaluada en m_a cumple:

$$\varphi(m_a) = \varphi(p_1)\cdots\varphi(p_a) = \prod_{i=1}^{a}(p_i - 1) \ .$$

Los conceptos anteriores permiten enunciar y demostrar las siguientes propiedades:

Proposición 23.

i) $\phi(x,a) = 1,$ para $0 < x \leq p_a$.

ii) $\phi(x,a) = \phi(x,a-1) - \phi\left(\dfrac{x}{p_a}, a-1\right)$.

iii) $\phi(m_a,a) = \varphi(m_a) = \prod_{i=1}^{a}(p_i - 1)$.

iv) $\phi(s \cdot m_a + t, a) = s \cdot \varphi(m_a) + \phi(t,a), \quad \forall t \in [0, m_a),\ s \geq 0$.

v) $\phi(t,a) = \varphi(m_a) - \phi(m_a - t - 1, a), \quad \forall t \in [0, m_a)$.

Demostración.

i) Para $0 < x \leq p_a$, solo $n = 1$ pertenece a $\{n \leq x;\ p|n \Rightarrow p > p_a\}$, luego por definición $\phi(x,a) = 1$.

ii) Los enteros en $[1,x]$ que no son divisibles por los primeros $a-1$ primos pueden separarse en aquellos que son divisibles por p_a y aquellos que no. Los del primer grupo no son divisibles por ninguno de los $a-1$ primeros pero sí por p_a, entonces son elementos de la forma $n \cdot p_a \leq x$ tal que $p|n \Rightarrow p > p_{a-1}$. Por la definición de ϕ, hay exactamente $\phi(x/p_a, a-1)$ elementos de este tipo.
Del mismo modo, como los del segundo grupo no son divisbles por p_a, estos son los enteros en $[1,x]$ que no son divisibles por los primeros a primos, de los cuales hay exactamene $\phi(x,a)$. Esto es,

$$\phi(x,a-1) = \phi\left(\frac{x}{p_a}, a-1\right) + \phi(x,a) \ .$$

iii) Por la Proposición 14,

$$\phi(m_a,a) = m_a - \sum_{i\leq a}\left\lfloor \frac{m_a}{p_i}\right\rfloor + \sum_{i<j\leq a}\left\lfloor \frac{m_a}{p_ip_j}\right\rfloor - \cdots + (-1)^a\left\lfloor \frac{m_a}{p_1\cdots p_a}\right\rfloor =$$

pero por definición de m_a, su cociente por cualquier producto de primos distintos es entero, luego

$$= m_a - \sum_{i\leq a}\frac{m_a}{p_i} + \sum_{i<j\leq a}\frac{m_a}{p_ip_j} - \cdots + (-1)^a\frac{m_a}{p_1\cdots p_a} =$$

y sacando factor común m_a y completando los productos, se obtiene

$$= m_a\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_a}\right) = \prod_{i=1}^{a}(p_i-1) = \varphi(m_a) \ .$$

iv) Por la Proposición 14

$$\phi(s\cdot m_a+t,a) = s\cdot m_a+t - \sum_{i\leq a}\left\lfloor \frac{s\cdot m_a+t}{p_i}\right\rfloor + \sum_{i<j\leq a}\left\lfloor \frac{s\cdot m_a+t}{p_ip_j}\right\rfloor - \cdots + (-1)^a\left\lfloor \frac{s\cdot m_a+t}{p_1\cdots p_a}\right\rfloor =$$

y por definición de m_a, el cociente de $s\cdot m_a$ por cualquier producto de primos distintos menores que p_a es entero, mientras que el de t no necesariamente lo cumple. Por el Lema 8.ii) se deduce

$$\begin{aligned} &= s\cdot m_a - \sum_{i\leq a}\frac{s\cdot m_a}{p_i} + \sum_{i<j\leq a}\frac{s\cdot m_a}{p_ip_j} - \cdots + (-1)^a\frac{s\cdot m_a}{p_1\cdots p_a} + \\ &+ t - \sum_{i\leq a}\left\lfloor\frac{t}{p_i}\right\rfloor + \sum_{i<j\leq a}\left\lfloor\frac{t}{p_ip_j}\right\rfloor - \cdots + (-1)^a\left\lfloor\frac{t}{p_1\cdots p_a}\right\rfloor = \\ &= s\cdot\phi(m_a,a) + \phi(t,a) = s\cdot\varphi(m_a) + \phi(t,a) \ . \end{aligned}$$

v) Por definición, $\phi(m_a,a)$ es la cantidad de enteros en $[1,m_a]$ no divisibles por los primeros a primos. De hecho, como 0 es divisible por cualquier entero, puede expresarse $\phi(x,a)$ como la cantidad de enteros en $[0,m_a]$ no divisibles por los primeros a primos. Estos valores, fijando $t \in [0,m_a)$, pueden separarse según sean $\leq t$ o $> t$. Por definición, hay $\phi(t,a)$ del primer tipo. Ahora bien, sea x del segundo tipo, por ser $m_a = p_1\cdots p_a$, se observa p_i con $i \leq a$ divide a x si y solo si divide a $m_a - x$; es decir, x no es divisible por los primeros a primos si y solo si no lo es $m_a - x$. Como x satisface

$$t < x \leq m_a \Rightarrow m_a - t > m_a - x \geq m_a - m_a = 0 \Rightarrow m_a - t - 1 \geq m_a - x \geq 0 \ ,$$

se deduce que habrá exactamente $\phi(m_a - t - 1, a)$ enteros del segundo tipo, luego

$$\phi(m_a,a) = \varphi(m_a) = \phi(t,a) + \phi(m_a - t - 1, a) \ . \qquad \square$$

La Proposición 23.ii) muestra la naturaleza recursiva de $\phi(x,a)$, que podría haberse intuido de las Proposiciones 16 y 17. La Proposición 23.iv) permite calcular $\phi(x,a)$ para cualquier x conociendo solamente el valor de $\phi(t,a)$ para $t \in [0,m_a)$. Es más, usada junto la Proposición 23.v) permite calcular $\phi(x,a)$ para cualquier x conociendo solamente el valor de $\phi(t,a)$ para $t \in [0,m_a/2)$.

Observación. Se recuerda que por definición, $\phi(x,a)$ cuenta los enteros positivos $\leq x$ que no son divisibles por ninguno de los a primeros primos. De este modo, si en una tabla con los enteros en $[1,m_a/2]$ se eliminan los múltiplos de los primeros a primos, el valor de $\phi(t,a)$ para $t \leq m_a/2$ será la cantidad de elementos no eliminados menores o iguales que t. Este procedimiento puede llevarse a cabo de forma eficiente modificando la criba de Eratóstenes para que elimine los múltiplos de solamente los a primeros primos en el intervalo $[1,m_a/2]$.

Si que se conocen los primeros a primos, se calcula $\phi(x,a)$ para todo $x \geq 0$ del modo siguiente:

1. Se calcula m_a y $m_a/2$.
2. Se crea una tabla crítica donde se almacenan los valores $\phi(t,a)$ para $t \leq m_a/2$; esto es, una tabla donde ϕ se registra solamente cuando cambia de valor. Para ello se usa el cribado mencionado en la Observación anterior, y cada elemento no eliminado se almacena junto con su posición.
3. Se calcula $\varphi(m_a)$ usando el Corolario 22.
4. Para $t \leq m_a/2$, se puede consultar el valor de $\phi(t,a)$ en la tabla crítica.
5. Para todo $t > m_a/2$, puede obtenerse $\phi(t,a)$ aplicando las Proposiciones 23.iv) y v).

Ejemplo. Para $a = 4$, se calcula $m_4 = p_1 \cdots p_4 = 2 \cdot 3 \cdot 5 \cdot 7 = 210,\ \ m_4/2 = 105$. La tabla crítica resulta:

Tabla crítica de $\phi(x,4)$											
1	1	19	5	37	9	53	13	71	17	89	21
11	2	23	6	41	10	59	14	73	18	97	22
13	3	29	7	43	11	31	15	79	19	101	23
17	4	31	8	47	12	67	16	83	20	103	24

Esto significa que, por ejemplo, $\phi(x,4) = 5 \quad \forall x$ con $19 \leq x < 23$.

Ahora, al ser $\varphi(m_a) = (2-1)(3-1)(5-1)(7-1) = 48$, para calcular $\phi(x,4)$ para un $x > m_4/2 = 105$ basta utilizar las Proposiciones 23.iv) y v) si es necesario:

$$\phi(200,4) = \varphi(m_4) - \phi(m_4 - 200 - 1, 4) = 48 - \phi(9,4) = 48 - 1 = 47$$
$$\phi(410,4) = \phi(m_a + 200, 4) = \varphi(m_4) + \phi(200,4) = 48 + 47 = 95$$

El tamaño de la tabla crítica de la función $\phi(x,a)$ crece considerablemente con el valor de a. Por esta razón, este procedimiento sólo se puede utilizar para calcular la suma de Legendre para valores de a relativamente pequeños.

Concretamente, en nuestra implementación del método de Mapes limitamos su uso para $a \leq 9$. El siguiente Corolario (del Teorema 20) permite calcular el valor de $\phi(x,a)$ de forma casi inmediata, siempre que se disponga de una tabla de primos suficientemente extensa.

Corolario 24 (Fórmula inversa de Lehmer)**.** Para $p_a < x < p_{a+1}^4$

$$\phi(x,a) = 1 + \pi(x) - a + P_2(x,a) + P_3(x,a)\ .$$

Además, cabe tener en cuenta que por la Proposición 23.i),

$$\phi(x,a) = 1,\ \text{para}\ 0 < x \leq p_a\ .$$

Nuestra implementación del código incluye en la clase *Mapes* la función *phixa* que permite calcular $\phi(x,a)$ usando, cuando sea posible, las tablas de ϕ construidas mediante la clase *Phi* o usando la función inversa de Lehmer, implementada en la función *phiLehmer* (ver Apéndices B.1 y B.3).

2.5. Método de Mapes

A partir de una reordenación de los términos de $\phi(x,a)$ mediante los valores $P_k(x,a)$, los métodos de Meissel y Lehmer consiguen reducir el valor de a necesario para calcular $\phi(x,a)$, mejorando así el tiempo de cálculo de $\pi(x)$. El método de Mapes, igual que estos dos métodos, se basa en una reordenación de los términos de $\phi(x,a)$. La diferencia fundamental con éstos es que el método de Mapes parte de la fórmula de Legendre; esto es, $a=\pi(\sqrt{x})$ y consigue facilitar el cálculo de $\phi(x,a)$ a base de expresarlo de forma iterada como funciones ϕ con valores de a sucesivamente más pequeños, explotando así la naturaleza recursiva de la función ϕ.

Mapes construye su método asociando unívocamente cada término de $\phi(x,a)$ con un índice $k\in[0,2^a)$.

Definición 7. Dado $a\geq 1$, consideremos la expresión binaria de los enteros $0\leq k<2^a$; esto es,

$$k=2^{a-1}\beta_{a-1}+2^{a-2}\beta_{a-2}+\cdots+2^1\beta_1+2^0\beta_0 \ \text{, donde } \beta_i\in\{0,1\}\ \forall i\ .$$

Entonces, se define

$$T_k(x,a)=(-1)^{\beta_0+\beta_1+\cdots+\beta_{a-1}}\left\lfloor\frac{x}{p_1^{\beta_0}p_2^{\beta_1}\cdots p_a^{\beta_{a-1}}}\right\rfloor\ .$$

Los términos $T_k(x,a)$ definen una reordenación de $\phi(x,a)$. La idea principal radica en asociar el n-ésimo primo p_n con la potencia 2^{n-1}. Además, el producto de primos distintos se asocia con la suma de las potencias correspondientes. Por ejemplo,

- 2 se asocia con 2^0, 3 con 2^1, 5 con 2^2, 7 con 2^3, etc.
- $2\cdot 5$ se asocia con 2^0+2^2, $2\cdot 3\cdot 7$ con $2^0+2^1+2^3$, etc.

Además, el signo de los términos en $\phi(x,a)$ está determinado por la paridad de la cantidad de primos en el denominador, lo que es equivalente a la elección de signo en cada $T_k(x,a)$. Por ello, se deduce que:

Proposición 25.

$$\phi(x,a)=\sum_{k=0}^{2^a-1}T_k(x,a)\ .$$

Ejemplo. Para $a=2$ puede calcularse $\phi(x,2)$ mediante la Proposición 14 (siendo $p_1=2, p_2=3$).

$$\phi(x,2)=x-\left\lfloor\frac{x}{p_1}\right\rfloor-\left\lfloor\frac{x}{p_2}\right\rfloor+\left\lfloor\frac{x}{p_1p_2}\right\rfloor\ .$$

Por otro lado, a partir de las expresiones binarias de $k=0,1,2,3=2^a-1$, los valores $T_k(x,a)$ resultan:

- $0=2^1\cdot 0+2^0\cdot 0\implies T_0(x,2)=(-1)^{0+0}\left\lfloor\frac{x}{p_1^0\cdot p_2^0}\right\rfloor=\lfloor x\rfloor=x\ .$
- $1=2^1\cdot 0+2^0\cdot 1\implies T_1(x,2)=(-1)^{1+0}\left\lfloor\frac{x}{p_1^1\cdot p_2^0}\right\rfloor=-\left\lfloor\frac{x}{p_1}\right\rfloor\ .$
- $2=2^1\cdot 1+2^0\cdot 0\implies T_2(x,2)=(-1)^{0+1}\left\lfloor\frac{x}{p_1^0\cdot p_2^1}\right\rfloor=-\left\lfloor\frac{x}{p_2}\right\rfloor\ .$
- $3=2^1\cdot 1+2^0\cdot 1\implies T_3(x,2)=(-1)^{1+1}\left\lfloor\frac{x}{p_1^1\cdot p_2^1}\right\rfloor=\left\lfloor\frac{x}{p_1p_2}\right\rfloor\ .$

Por lo que se cumple $\phi(x,2)=\sum_{k=0}^{3}T_k(x,2)$.

Notación. Escribiremos $p^a||b$ si p^a es la mayor potencia de p que divide a b; esto es, $p^a|b$ pero $p^{a+1}\nmid b$.

Definición 8. Sea M un entero tal que $M < 2^a$, y sea i tal que $2^i || M$, se define

$$\gamma(M,x,a) = \sum_{k=M}^{M+2^i-1} T_k(x,a) \ .$$

Además, las agrupaciones $\gamma(M,x,a)$ que se han definido también forman una reordenación de los términos de $\phi(x,a)$ eligiendo los valores de M apropiados.

Proposición 26.

$$\phi(x,a) = T_0(x,a) + \gamma(2^0,x,a) + \gamma(2^1,x,a) + \cdots + \gamma(2^{a-1},x,a) \ .$$

Demostración. Basta desarrollar los términos $\gamma(2^i,x,a)$ y aplicar la Proposición 25.

$$T_0(x,a) + \gamma(2^0,x,a) + \gamma(2^1,x,a) + \cdots + \gamma(2^{a-1},x,a) =$$

$$= T_0(x,a) + \sum_{k=2^0}^{2^0+2^0-1} T_k(x,a) + \sum_{k=2^1}^{2^1+2^1-1} T_k(x,a) + \cdots + \sum_{k=2^{a-1}}^{2^{a-1}+2^{a-1}-1} T_k(x,a) =$$

$$= T_0(x,a) + \sum_{k=1}^{1} T_k(x,a) + \sum_{k=2}^{3} T_k(x,a) + \cdots + \sum_{k=2^{a-1}}^{2^a-1} T_k(x,a) =$$

$$= \sum_{k=0}^{2^a-1} T_k(x,a) = \phi(x,a) \ .$$

□

Nota. De forma similar a ϕ, conviene extender el dominio de $T_k(x,a)$, como función de x, a $\mathbb{Z}$ tomando:

$$T_k(0,a) = 0 \ ,$$
$$T_k(x,a) = -T_k(-x,a), \quad \forall x < 0 \ .$$

Observación. Para $k < 2^a$, si $2^i | k$ debe cumplirse $\beta_j = 0 \quad \forall j < i$. De este modo,

$$T_k(x,a) = (-1)^{\beta_i + \beta_{i+1} + \cdots + \beta_{a-1}} \left\lfloor \frac{x}{p_{i+1}^{\beta_i} p_{i+2}^{\beta_{i+1}} \cdots p_a^{\beta_{a-1}}} \right\rfloor \ .$$

Por la naturaleza recursiva de $\phi(x,a)$ y $P_k(x,a)$ es de esperar que los valores $T_k(x,a)$ tengan propiedades similares. Esto se muestra en el resultado siguiente.

Proposición 27. Si $2^i | k$, entonces $T_{k'}\big(T_k(x,a), i\big) = T_{k+k'}(x,a)$ para $k' < 2^i$.

Demostración. Puesto que $k' < 2^i$, su expresión binaria es

$$k' = 2^{i-1}\beta'_{i-1} + 2^{i-2}\beta'_{i-2} + \cdots + 2^1\beta'_1 + 2^0\beta'_0 \ .$$

Entonces,

$$T_{k'}\big(T_k(x,a), i\big) = (-1)^{\beta'_0 + \beta'_1 + \cdots + \beta'_{i-1}} \left\lfloor \frac{T_k(x,a)}{p_1^{\beta'_0} p_2^{\beta'_1} \cdots p_i^{\beta'_{i-1}}} \right\rfloor =$$

$$= (-1)^{\beta'_0 + \beta'_1 + \cdots + \beta'_{i-1}} (-1)^{\beta_i + \beta_{i+1} + \cdots + \beta_{a-1}} \left\lfloor \frac{\left\lfloor x / (p_{i+1}^{\beta_i} p_{i+2}^{\beta_{i+1}} \cdots p_a^{\beta_{a-1}}) \right\rfloor}{p_1^{\beta'_0} p_2^{\beta'_1} \cdots p_i^{\beta'_{i-1}}} \right\rfloor =$$

y aplicando el Lema 8.iii)

$$= (-1)^{\beta'_0+\beta'_1+\cdots+\beta'_{i-1}+\beta_i+\beta_{i+1}+\cdots+\beta_{a-1}} \left\lfloor \frac{x}{p_1^{\beta'_0} p_2^{\beta'_1} \cdots p_i^{\beta'_{i-1}} p_{i+1}^{\beta_i} p_{i+2}^{\beta_{i+1}} \cdots p_a^{\beta_{a-1}}} \right\rfloor =$$

$$= T_{k+k'}(x,a) \ .$$

□

Además, los términos $T_M(x,a)$ y las agrupaciones $\gamma(M,x,a)$ se relacionan a través de la función ϕ mediante la siguiente forma, lo que permitirá descomponer sucesivamente esta función usando las dos Proposiciones anteriores.

Proposición 28. Si $2^i|M$, entonces $\phi(T_M(x,a),i) = \gamma(M,x,a)$.

Demostración. Por la Proposición 25, puede ponerse

$$\phi(T_M(x,a),i) = \sum_{k=0}^{2^i-1} T_k\big(T_M(x,a),i\big) \ .$$

Ahora bien, como $2^i|M$ y $k < 2^i$, por la Proposición 27

$$\sum_{k=0}^{2^i-1} T_k\big(T_M(x,a),i\big) = \sum_{k=0}^{2^i-1} T_{k+M}(x,a) \ .$$

Tomando $k' = k+M (\Rightarrow k' = M \Leftrightarrow k = 0)$ para desplazar el índice en la suma, se tiene

$$\sum_{k=0}^{2^i-1} T_{k+M}(x,a) = \sum_{k'=M}^{M+2^i-1} T_{k'}(x,a) \ .$$

Finalmente, por la Definición 8 se demuestra el enunciado.

$$\sum_{k'=M}^{M+2^i-1} T_{k'}(x,a) = \gamma(M,x,a) \ .$$

□

Corolario 29.

i) $\phi(x,a) = T_0(x,a) + \phi(T_{2^0}(x,a),0) + \phi(T_{2^1}(x,a),1) + \cdots + \phi(T_{2^{a-1}}(x,a),a-1)$.

ii) Si $2^i|M$,

$$\phi(T_M(x,a),i) = T_M(x,a) + \phi(T_{M+2^0}(x,a),0) + \phi(T_{M+2^1}(x,a),1) + \cdots + \phi(T_{M+2^{i-1}}(x,a),i-1) \ .$$

Demostración.

i) Basta tomar la expresión de $\phi(x,a)$ dada en la Proposición 26:

$$\phi(x,a) = T_0(x,a) + \gamma(2^0,x,a) + \gamma(2^1,x,a) + \cdots + \gamma(2^{a-1},x,a) \ ,$$

y por la Proposición 28, $\gamma(2^i,x,a) = \phi(T_{2^i}(x,a),i)$, obteniendo el resultado.

ii) Desarrollando $\phi(T_M(x,a),i)$ usando el apartado i), resulta

$$\phi(T_M(x,a),i) = T_0\left(T_M(x,a),i\right) + \phi\left(T_{2^0}\left(T_M(x,a),i\right),0\right) + \phi\left(T_{2^1}\left(T_M(x,a),i\right),1\right) + \cdots +$$
$$+\phi\left(T_{2^{i-1}}\left(T_M(x,a),i\right),i-1\right) .$$

Ahora bien, como $2^i|M$ y $2^j < 2^i \quad \forall j \in \{0,\ldots,i-1\}$, puede aplicarse la Proposición 27, demostrando el enunciado:

$$\phi(T_M(x,a),i) = T_M(x,a) + \phi(T_{M+2^0}(x,a),0) + \phi(T_{M+2^1}(x,a),1) + \cdots + \phi(T_{M+2^{i-1}}(x,a),i-1) .$$

□

El esquema a seguir para aplicar el método de Mapes es el siguiente:

1. Se crea una tabla de primos, cuanto más grande mejor, usando la criba parcial de Eratóstenes.
2. Se crean las tablas críticas de $\phi(x,a)$. Un valor máximo razonable es $a = 9$.
3. La función $\phi(x,a)$ se descompone aplicando el Corolario 29.i)
4. Cada sumando se intenta calcular usando las tablas críticas de ϕ. Si no es posible, se intenta calcular usando la fórmula inversa de Lehmer. Si no es posible, se descompone usando el Corolario 29.ii) y a cada nuevo término se le vuelve a aplicar este paso.
5. Habiendo calculado $\phi(x,a)$, el valor de $\pi(x)$ viene dado por la fórmula de Legendre

$$\pi(x) = a - 1 + \phi(x,a), \ \text{ con } a = \pi\left(\sqrt{x}\right) .$$

Ejemplo. Cálculo de $\pi(400)$ mediante el método de Mapes.

Para aplicar el método de Mapes es necesario tener una tabla con cierta cantidad de números primos. Para este ejemplo, se tomará una tabla con los primos en el intervalo $[1,25]$, obtenidos mediante la criba de Eratóstenes. Además, se dispone de las tablas críticas de $\phi(x,a)$ para todo $a \leq 4$.

Puesto que $\sqrt{400} = 20$, entonces $a = \pi(20) = 8$, y por la fórmula de Legendre,

$$\pi(400) = 8 - 1 - \phi(400,8) .$$

Para calcular $\phi(400,8)$, se aplica el Corolario 29.i), obteniendo

$$\begin{aligned}\phi(400,8) = T_0(400,8) &+ \phi(T_{2^0}(400,8),0) + \phi(T_{2^1}(400,8),1) + \phi(T_{2^2}(400,8),2) + \phi(T_{2^3}(400,8),3) + \\ &+ \phi(T_{2^4}(400,8),4) + \phi(T_{2^5}(400,8),5) + \phi(T_{2^6}(400,8),6) + \phi(T_{2^7}(400,8),7) .\end{aligned}$$

Se observa que $T_0(400,8) = 400$, y que una vez calculado $T_{2^i}(400,8)$ a partir de la Definición 7 los valores $\phi(T_{2^i}(400,8),i)$ para todo $i \leq 4$ pueden obtenerse directamente a través de las tablas críticas de ϕ correspondientes. De este modo, se tiene:

$$\begin{aligned}\phi(400,8) &= 400 + \phi(-200,0) + \phi(-133,1) + \phi(-80,2) + \phi(-57,3) + \phi(-36,4) + \\ &+ \phi(T_{2^5}(400,8),5) + \phi(T_{2^6}(400,8),6) + \phi(T_{2^7}(400,8),7) = \\ &= 400 - 200 - 67 - 27 - 15 - 8 + \\ &+ \phi(T_{2^5}(400,8),5) + \phi(T_{2^6}(400,8),6) + \phi(T_{2^7}(400,8),7) .\end{aligned}$$

El resto de valores no pueden calcularse usando las tablas críticas, pero quizá si puedan calcularse con la fórmula de Lehmer inversa, dada por

$$\phi(x,a) = 1 + \pi(x) - a + P_2(x,a) + P_3(x,a), \ \text{ si } p_a < x < p_{a+1}^4 .$$

Al ser $T_{2^5}(400,8) = -30$, $T_{2^6}(400,8) = -23$, $T_{2^7}(400,8) = -21$, no se podrá calcular $\phi(T_{2^5}(400,8),5)$ porque no se puede calcular $\pi(30)$, ya que solo se dispone de una tabla con los primos en $[1,25]$. En cambio, $\pi(23)$ y $\pi(21)$ podrían obtenerse mediante una búsqueda binaria en la tabla de primos.

Como $p_6 = 13 < 23 < p_7^4 = 83.521$ y $p_7 = 17 < 21 < p_8^4 = 130.321$, ambos cumplen la condición de la fórmula de Lehmer inversa, luego puede calcularse $\phi(T_{2^6}(400,8),6) = -4$, $\phi(T_{2^7}(400,8),7) = -2$.

De este modo, se tiene:

$$\phi(400,8) = 77 + \phi(T_{2^5}(400,8),5) \ .$$

Para calcular $\phi(T_{2^5}(400,8),5)$ se aplica el Corolario 29.ii), obteniendo

$$\begin{aligned}\phi(T_{2^5}(400,8),5) = T_{2^5}(400,8) + \phi(T_{2^5+2^0}(400,8),0) + \phi(T_{2^5+2^1}(400,8),1) + \phi(T_{2^5+2^2}400,8),2) + \\ + \phi(T_{2^5+2^3}(400,8),3) + \phi(T_{2^5+2^4}(400,8),4) \ .\end{aligned}$$

Igual que antes, los valores $T_{2^5+2^i}(400,8)$ pueden calcularse aplicando la Definición 7, y para $i \leq 4$ los valores $\phi(T_{2^5+2^i}(400,8),i)$ pueden obtenerse su valor mediante las tablas críticas de ϕ.

De este modo resulta $\phi(T_{2^5}(400,8),5) = -30+15+5+2+1+1 = -6$, y así $\phi(400,8) = 77-6 = 71$, por lo que

$$\pi(400) = 8 - 1 + 71 = 78 \ .$$

A la hora de implementar el método es útil seguir el siguiente esquema, donde se calculan los términos de la descomposición de $\phi(x,a)$ uno a uno siguiendo el método descrito antes, acumulando los valores a medida que se consigan calcular en una variable llamada y.

1. Se inicializan los siguientes valores: $y = 0$, $M = 0$, $i = a = \pi(\sqrt{x})$.

2. Se calcula el valor $T_M(x,a)$.

3. Si se puede calcular $\phi(T_M(x,a),i)$ mediante alguna tabla crítica de ϕ o la fórmula inversa de Lehmer, se calcula, se toma $y = y + \phi(T_M(x,a),i)$, $M = M + 2^i$ y se establece i tal que $2^i || M$.
   Si no puede calcularse, se toma $y = y + T_M(x,a)$, $M = M + 2^0$ y se establece i tal que $2^i || M$.
   Si no se cumple $M = 2^a$ se vuelve al paso 2.

4. $\pi(x) = a - 1 + y$.

Nota. En [16] y en [12] se menciona que el coste en tiempo del método de Mapes es de orden $O(x^{0,7})$. Mediante nuestra implementación del método de Mapes (ver Apéndice B) se ha generado una tabla de primos (ver Apéndice A) con sus tiempos de ejecución. Expresando cada tiempo de ejecución del cálculo de $\pi(x)$ de la forma x^b, se ha obtenido que para todo x, los valores b se ajustan a $0{,}7 \pm 0{,}185$.

Bibliografía

[1] APOSTOL, T. M., *Introduction to Analytic Number Theory*, New York, NY.: Springer, 1976.

[2] BOHMAN, J., *On the Number of Primes Less Than a Given Limit*, Nordsik Tidskrift för Informationsbehandling (BIT) 12:576-577, 1972.

[3] CHEBYSHEV, P.L., *Sur la fonction qui détermine la totalité des nombres premiers inferieurs à une limite donée*, Mem. Ac. Sc. St. Pétersbourg, 6: 141-157, 1851.

[4] DELEGLISE, M., RIVAT, J., *Computing $\pi(x)$: The Meissel, Lehmer, Lagarias, Miller, Odlyzko Method*, Math. Comp., 65:235-245, 1996.

[5] EULER, L., *Variae observationes circa series infinitas*, Commentarii Academiae Scientiarum lmperialis Petropolitanae, 9: 160-188, 1737. [*Opera Omnia* (1), 14; 216-244].

[6] GAUSS, C.F., *Letter to Encke*, con fecha de 24 de Diciembre, 1849. [*Werke*, Vol. II, 444-447].

[7] HADAMARD, J., *Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques.*, Bull. Soc, Math. France, 24:199-220, 1798.

[8] LAGARIAS, J.C., MILLER, V.S., ODLYZKO, A.M., *Computing $\pi(x)$: The Meissel-Lehmer Method*, Math. Comp., 44:537-560, 1798.

[9] LEGENDRE, A.M., *Essai sur la Théorie des Nombres*, Paris: Duprat, 1798.

[10] LEGENDRE, A.M., *Théorie des Nombres*, Tercera Edición, Paris, Vol.2:65, 1830.

[11] LEHMER, D.H., *On the Exact Number of Primes Less Than a Given Limit*, Ill. Journ. Math. 3:381-388, 1959.

[12] MAPES, D.C., *Fast Method for Computing the Number of Primes Less than a Given Limit*, Math. Comp., 17:179-185, 1963.

[13] MEISSEL, E.D.F., *Über die Bestimmung der Primzahlenmenge innerhalb gegebener Grenzen*, Math. Ann., 2:636-642, 1870.

[14] NARKIEWICZ, N., *The Development of Prime Number Theory, From Euclid to Hardy and Littlewood*, Boston, Mass.: Birkhäuser, 1994.

[15] RIEMANN, B., *Über die Anzahl der Primzahlen unter einer gegebener Grösse*, Monatsber. Alad. Berlín, 671-680, 1859.

[16] RIESEL, H., *Prime numbers and computer methods for factorization*, Berlin: Springer, 2000.

[17] VALLÉE-POUSSIN, CH. DE LA, *Recherches analytiques sur la théorie des nombres premiers*, Ann. Soc. Sci. Bruxelles, 20:183-256, 281,297, 1896.

Apéndice A

Tabla de Primos

La siguiente tabla muestra los valores de $\pi(x)$ y su tiempo de ejecución en nanosegundos obtenidos mediante el método de Mapes implementado en el Apéndice B.

x	$\pi(x)$	Tiempo	x	$\pi(x)$	Tiempo	x	$\pi(x)$	Tiempo
10^6	78,498	58300	10^9	50,847,534	44000	10^{12}	37,607,912,018	436304300
$2\cdot10^6$	148,933	61800	$2\cdot10^9$	98,222,287	10376100	$2\cdot10^{12}$	73,301,896,139	961608700
$3\cdot10^6$	216,816	83700	$3\cdot10^9$	144,449,537	8244600	$3\cdot10^{12}$	108,340,298,703	1506498700
$4\cdot10^6$	283,146	59900	$4\cdot10^9$	189,961,812	8979100	$4\cdot10^{12}$	142,966,208,126	2089141100
$5\cdot10^6$	348,513	73700	$5\cdot10^9$	234,954,223	9425800	$5\cdot10^{12}$	177,291,661,649	2704715900
$6\cdot10^6$	412,849	61600	$6\cdot10^9$	279,545,368	10377600	$6\cdot10^{12}$	211,381,427,039	3328581700
$7\cdot10^6$	476,648	69300	$7\cdot10^9$	323,804,352	11294600	$7\cdot10^{12}$	245,277,688,804	3994422000
$8\cdot10^6$	539,777	61000	$8\cdot10^9$	367,783,654	10951900	$8\cdot10^{12}$	279,010,070,811	4705300400
$9\cdot10^6$	602,489	61400	$9\cdot10^9$	411,523,195	12255800	$9\cdot10^{12}$	312,600,354,108	5412416600
10^7	664,579	57200	10^{10}	455,052,511	13482100	10^{13}	346,065,536,839	6269304800
$2\cdot10^7$	1,270,607	60900	$2\cdot10^{10}$	882,206,716	18802600	$2\cdot10^{13}$	675,895,909,271	16892808700
$3\cdot10^7$	1,857,859	60900	$3\cdot10^{10}$	1,300,005,926	24136900	$3\cdot10^{13}$	1,000,121,668,853	33682266700
$4\cdot10^7$	2,433,654	74000	$4\cdot10^{10}$	1,711,955,433	29904500	$4\cdot10^{13}$	1,320,811,971,702	57576102300
$5\cdot10^7$	3,001,134	58500	$5\cdot10^{10}$	2,119,654,578	35570500	$5\cdot10^{13}$	1,638,923,764,567	92281808300
$6\cdot10^7$	3,562,115	65300	$6\cdot10^{10}$	2,524,038,155	42181400	$6\cdot10^{13}$	1,955,010,428,258	137696428800
$7\cdot10^7$	4,118,064	83400	$7\cdot10^{10}$	2,925,699,539	42547900	$7\cdot10^{13}$	2,269,432,871,304	192743130800
$8\cdot10^7$	4,669,382	62100	$8\cdot10^{10}$	3,325,059,246	45118700	$8\cdot10^{13}$	2,582,444,113,487	265886747700
$9\cdot10^7$	5,216,954	51300	$9\cdot10^{10}$	3,722,428,991	49176600	$9\cdot10^{13}$	2,894,232,250,783	351198079000
10^8	5,761,455	77800	10^{11}	4,118,054,813	54030700	10^{14}	3,204,941,750,802	455220287000
$2\cdot10^8$	11,078,937	71700	$2\cdot10^{11}$	8,007,105,059	83884200			
$3\cdot10^8$	16,252,325	55100	$3\cdot10^{11}$	11,818,439,135	116589400			
$4\cdot10^8$	21,336,326	51200	$4\cdot10^{11}$	15,581,005,657	153798900			
$5\cdot10^8$	26,355,867	57800	$5\cdot10^{11}$	19,308,136,142	197247100			
$6\cdot10^8$	31,324,703	49800	$6\cdot10^{11}$	23,007,501,786	244782400			
$7\cdot10^8$	36,252,931	44700	$7\cdot10^{11}$	26,684,074,310	293224900			
$8\cdot10^8$	41,146,179	46600	$8\cdot10^{11}$	30,341,383,527	337815900			
$9\cdot10^8$	46,009,215	47000	$9\cdot10^{11}$	33,981,987,586	383936000			

Apéndice B

Código del Método de Mapes

Puede accederse al proyecto completo desde GitHub[1].

B.1. Clase *Phi*

```java
/**
 * La clase Phi genera las tablas críticas de phi(x,a) para un a
 * fijo en cada una de sus instancias.
 * Además, (para cada a) el método valueAtX() obtiene el valor
 * phi(x,a) para todo x usando las tablas críticas o la
 * Proposición 23.iv) y el Corolario 24.
 *
 * @author ppita
 */
public class Phi {

    //El entero a es fijo para cada instancia de Phi.
    private final int a;


    /* Se incializan los siguientes valores:
     * El entero ma viene dado por m_a=p_1*p_2*...*p_a y ma2=ma/2.
     * El entero phima es varphi(m_a)=(p_1-1)*(p_2-1)*...*(p_a-1).*/
    private long ma=1;
    private long ma2;
    private long phima=1;

    //El array table[] almacena la tabla crítica de phi para el a dado.
    private long[] table;


    /* El constructor de Phi fija el valor de a pasado como parámetro
     * y mediante generateTable() genera la tabla crítica table[]. */
    public Phi(int a){
        this.a=a;
        generateTable();
    }

    /**
     * Conociendo la tabla crítica de phi(x,a), para x menor que ma2
     * puede obtenerse phi(x,a) consultándola.
```

[1] https://github.com/PhilPF/Mapes-Method

```
     * En caso contrario se usa la Proposición 23.iv) y el Corolario 24.
     *
     * @param x
     * @return phi(x,a)
     */
    public long valueAtX(long x){
        long r=x%ma;
        long z=(x/ma)*phima;
        if (r<ma2){
            int c = Arrays.binarySearch(table, r+1);
            /* binarySearch() devuelve el índice de r+1 si aparce en la
             * tabla y pos:={-(posición donde debería aparecer) -1} en
             * caso contrario.*/
            z+=c<0?-c-1:c;
        }
        else {
            int c = Arrays.binarySearch(table, ma-r);
            /* binarySearch() devuelve el índice de ma-r si aparce en la
             * tabla y pos:={-(posición donde debería aparecer) -1}
             * en caso contrario.*/
            z+=c<0?phima+c+1:phima-c;
        }
        return z;
    }

    /* La función generateTable genera la tabla crítica de phi para a.
     * Además, define ma, ma2 y phima. */
    private void generateTable(){
        for (int i=0; i<a; i++){
            long p=Mapes.primes[i];
            ma*=p;
            phima*=(p-1);
        }
        ma2=ma/2;

        /* Se genera la tabla con la función Mapes.eratosthenesInterval(),
         * usando los primeros a primos para cribar el intervalo [a,ma2],
         * y se fuerza que el primer valor sea 1 para que la tabla
         * sea correcta.*/
        table = Mapes.eratosthenesInterval((int)Mapes.primes[a-1],ma2,
            Mapes.primes ,a);
        table[0]=1;
    }
}
```

B.2. Clase *BigInt*

```java
/**
 * La clase BigInt está diseñada para representar los números M del
 * valor T_M(x,a), puesto que crecen con facilidad pero las operaciones
 * que se hacen con ellos son limitadas, por lo que esta clase ayuda
 * a manejarlos, almacenando su expresión binaria en un BitSet.
 * Además, se lleva la cuenta de los elementos no nulos que posee.
 *
 * @author ppita
 */

public class BigInt {

    private BitSet arrayM;
    private int a;
    private int maxDiv;
    private int count=0;

    /* El constructor de BigInt crea el arrayList arrayM de tamaño a+1.
     * Al crearlo todos sus elementos son nulos, luego es M=0.
     * Se inicializa maxDiv=0 (inicialmente el valor no es correcto,
     * pero se actualizará antes de usarse).*/
    public BigInt(int a){
        this.a=a;
        maxDiv=0;
        arrayM=new BitSet(a+1);
    }

    //Puede usarse para obtener el array en formato String
    @Override
    public String toString(){
        return arrayM.toString();
    }

    //Devuelve el número de elementos no nulos del array.
    public int getCount(){
        return count;
    }

    // Devuelve el valor almacenado en el índice i de arrayM.
    public boolean get(int i){
        return arrayM.get(i);
    }

    /**
    * La función add2Powi toma el valor i y hace M+=2^i.
    * El siguiente ejemplo ilustra el algoritmo:
    *  Ej: M= 2^3+2^4+2^7. Si se quiere sumar 2^3, entonces
    *       2^3+M= 2^3+2^3+2^4+2^7= 2*2^3+2^4+2^7= 2^4+2^4+2^7=
    *            = 2*2^4+2^7= 2^5+2^7
    *       Como 3 aparece en el array, se sabe que 2^3+2^3=2^4, luego
    *       el valor 3 se elimina y se repite el proceso con 4.
    *       Como 4 aparece en el array, idem. Se repite con 5.
    *       Ahora, como 5 no está en el array, se coloca en el.
    *
    * @param i Exponente de la potencia de 2 a sumar
    */
```

```
    public void add2Powi(int i){
        int j=i;
        //Se da valor 0 a todos los elementos no nulos siguiendo a i.
        for (j=i; j<=a && arrayM.get(j); j++){
            arrayM.clear(j);
            count--;
        }
        //Se pone 1 en el índice siguiente al último valor no nulo.
        arrayM.set(j);
        count++;

        //Además, se actualiza maxDiv del siguiente modo.
        if(i<=maxDiv) maxDiv=j;
    }

    //Se devuelve el exponente del máximo divisor de M.
    public int getMaxDiv(){
        return maxDiv;
    }

    //Se comprueba si M es exactamente de la forma M=2^a.
    public boolean isPowA(){
        return arrayM.get(a);
    }

}
```

B.3. Clase *Mapes*

```java
/**
 * La clase Mapes es la clase principal. Junto con las clases BigInt
 * y Phi, implementa el metodo de Mapes para calcular el número de
 * primos menores o iguales que x para valores de x dados.
 *
 * @author ppita
 */
public class Mapes {

    /* Se almacenará en el array primes los primos <= maxPrime
     * Conociendo pi(1e9)=50847534 puede crearse el array primes[] del
     * tamaño justo. Los valores que aprecen en bigPi[] son pi(ie8)
     * con i=2,...,10 y ayudan a simplificar los 'cortes' de la criba
     * parcial, aunque no son estrictamente necesarios.*/
    private static final int maxPrime = (int)1e9;
    public static long[] primes = new long[50847534];
    private static final int[] bigPi = {5761455, 11078937, 16252325,
       21336326, 26355867, 31324703, 36252931, 41146179, 46009215};

    /* Se almacenará en el array phiArray instancias de la clase Phi(a)
     * con 4<=a<=maxA.*/
    private static final int maxA = 9;
    private static Phi[] phiArray = new Phi[maxA+1];


    public static void main(String[] args){
        /* Se guarda en tInicial el tiempo inicial para poder llevar la
         * cuenta del tiempo de ejecución y se usarán t1Parcial,t2Parcial
         * para calcular el timepo de cada parte. */
        long tInicial = System.nanoTime();
        long t1Parcial, t2Parcial;

        /* Se generan los primos hasta 1e8 y se almacenan en primes[]
         * mediante la función eratosthenes().*/
        int partition = (int) 1e8;
        eratosthenes(primes, partition);

        /* Usando los valores ya obtenidos y almacenados en primes[], se
         * criban el resto de primos hasta 1e9 a intervalos de longitud
         * 1e8 mediante la función eratosthenesInterval().*/
        for(int i=0; i<bigPi.length; i++) eratosthenesInterval(primes,
           bigPi[i], (i+1)*partition,  (i+2)*partition, primes, (int)
           intSqrt(bigPi[i]));

        t1Parcial = System.currentTimeMillis();
        t2Parcial = t1Parcial;
        System.out.println("Tabla de primos generada en tiempo(ns):"+(
           t1Parcial-tInicial));

        // Se generan y se guardan las instancias de Phi(a) en phiArray.
        for (int i=4; i<=maxA; i++) phiArray[i] = new Phi(i);

        t1Parcial = System.nanoTime();
        System.out.println("Tablas de phi generadas en tiempo(ns):"+(
           t1Parcial-t2Parcial));
        t2Parcial = t1Parcial;
```

```
        System.out.println();

        /* Se generan los x para los cuales se calculará pi(x).
         * Son i*10^j para 1<=i<=9, pot10Inicial<=j<=pot10Final.*/
        int pot10Inicial = 5;
        int pot10Final = 14;
        double[] xArray = new double[(pot10Final-pot10Inicial+1)*9];
        for (int i=pot10Inicial; i<=pot10Final; i++){
            for (int j=1; j<=9; j++){
                xArray[(j-1)+9*(i-pot10Inicial)]=j*Math.pow(10, i);
            }
        }

        for (double xd : xArray){
            long x = (long) xd;

            //Se evalúa la función mapes en x y se devuelve su valor.
            long pix=mapes(x);

            t1Parcial = System.nanoTime();

            System.out.println("Valor de pi("+(double)x+") calculado en
               tiempo(ns):"+(t1Parcial-t2Parcial));
            t2Parcial = t1Parcial;

            System.out.println("pi("+(double)x+")="+pix);

            System.out.println();
        }

        t1Parcial = System.nanoTime()-tInicial;
        System.out.println("Programa finalizado en tiempo(ns):"+t1Parcial);



    }

    // La función mapes() calcula pi(x) usando el Método de Mapes.
    public static long mapes(long x){

        /* Se inicializan las variables antes de la primera iteración.
         * y=0; M=0; i=a=pi(sqrt(x));*/
        long y=0;
        int a=(int) smallPi(intSqrt(x));
        int i=a;
        BigInt M = new BigInt(a);
        long T_M;
        long phi;

        // En el caso que M=2^a, se finaliza el cálulo de phi(x,a)
        while(!M.isPowA()){

            // Se calcula T_M(x,a).
            T_M=T(M,x,a);

            /* Se intenta calcular phi(T_M(x,a),i) con la función
             * phixa() mediante las tablas la fórmula inversa de Lehmer.
```

```java
             * Si no es posible, se recoge una excepción y se procede
             * según indica el algoritmo de Mapes. */
            try{
                phi = phixa(T_M,i);

                /* Como ha podido calcularse phi, se toma
                 * y+=phi(T_M(x,a),i), M+=2^i e i tal que 2^i|M.*/
                y+=phi;
                M.add2Powi(i);
                i=M.getMaxDiv();
            }catch (Exception exPhi){
                /* Al no poder calcularse phi, se toma
                 * y+=T_M(x,a), M+=1 e i tal que 2^i|M.*/
                M.add2Powi(0);
                i=M.getMaxDiv();
                y+=T_M;
            }
        }

        //Por último, se devuelve el valor pi(x)=a-1+phi(x,a)
        return y+a-1;

    }
    /* La función T recibe como parámetros los valores M,x,a y devuelve
     * T_M(x,a) calculado aplicando su definición.*/
    public static long T(BigInt M, long x, int a){

        /* Se inicializan los siguientes valores.
         * En betaSum se almacenará la suma de los coeficientes de la
         * expresión binaria de M; i.e, beta_0+beta_1+...+beta_{a-1}*/
        long betaSum=M.getCount();
        long T = x;

        /* Puesto que hay en M betaSum valores y el menor no nulo
         * es M.getMaxDiv(), el siguiente bucle garantiza acceder
         * a todos los valores no nulos en M.*/
        int counter=0;
        for (int i=M.getMaxDiv(); counter<betaSum; i++){
            //Por cada valor en i, se divide T por p_{i+1}.
            if(M.get(i)){
                counter++;
                T/=primes[i];
            }
        }

        //Además, el signo de T se determina con la paridad de betaSum.
        T*=betaSum%2==0?1:-1;

        return T;
    }

    /* La función phixa() recibe como parámetros los valores x, a y
     * devuelve phi(x,a). Si 0<=a<4, se calcula el valor directamente.
     * Si 4=<a<=maxA se calcula el valor mediante las tablas ceíticas.
     * Si maxA<a, se intenta calcular mediante la función phiLehmer().
     * Si no puede calcularse, se lanza una excepción. */
    public static long phixa(long x ,int a) throws Exception{
```

```
        //Se devuelve el valor del caso trivial
        if(x==0) return 0;

        //Para x<0 se sabe que phi(x,a)=-phi(-x,a)
        if(x<0) return -phixa(-x,a);

        //Para valores pequeños de a, puede calcularse directamente.
        if(a==0) return x;
        if(a==1) return (x+1)/2;
        if(a==2) return x-x/2-x/3+x/6;
        if(a==3) return x-x/2-x/3-x/5+x/6+x/10+x/15-x/30;

        //Si hay tabla, se calcula el valor con ella.
        if(a<=maxA) return phiArray[a].valueAtX(x);

        /* Las siguientes son las condiciones para aplicar la
         * fórmula inversa de Lehmer.*/
        if(primes[a-1]<x){
            if(x<Math.pow(primes[a], 4)){
                if(x<=maxPrime){
                    return phiLehmer(x,a);
                }
            }
        } else return 1;

        /* Si no se cumplen ninguna de las condiciones anteriores no
         * puede calcularse phi(x,a), luego se lanza una excepción. */
        throw new Exception("Phi Exception");

    }

    /* Esta función calcula phi(x,a) mediante la fórmula inversa
     * de Lehmer. Solo debe llamarse tras comprobarse que se cumplen
     * las condiciones necesarias. */
    public static long phiLehmer(long x, int a){

        // Se define pix=pi(x), b=pi(sqrt(x)), c=pi(cbrt(x)).
        long pix=smallPi(x);
        long b=smallPi(intSqrt(x));
        long c=smallPi(intCbrt(x));

        /* Como phi(x,a)=pi(x)-a+1+P2(x,a)+P3(x,a), se inicializa sum a
         * pi(x)-a+1 y se añadirán los términos restantes. */
        long sum=pix-a+1;

        /* Se calculan los valores de P2(x,a) y P3(x,a) de forma
         * conjunta y se añaden a sum.
         * La expresión siguiente se basa en las fórmulas:
         * P2(x,a)=sum_{a<i<=b}{pi(x/p_i)-i+1}
         * P3(x,a)=sum_{a<i<=c}sum_{1<=j<=bi}{pi(x/{p_i*p_j})-j+1}*/
        for (int i=a+1; i<=b; i++){
            long w=x/primes[i-1];
            sum+=(smallPi(w)-i+1);
            if(i<=c){
                long bi=smallPi(intSqrt((int)w));
                for (int j=i; j<=bi; j++){
                    sum+=(smallPi(w/primes[j-1])-j+1);
                }
```

```
            }
        }
        return sum;
    }

    /**
     * La función smallPi recibe el valor x y devuelve la cantidad de
     * primos menores o iguales que x que se hallan en la tabla primes
     * mediante una búsqueda binaria.
     * La función binarySearch() devuelve el índice de x si x es primo
     * por lo que en este caso se le suma 1 por empezar el array en 0.
     * En otro caso, devuelve
     * pos:={-(posición donde debería aparecer) -1},
     * por lo que se quiere -pos-1.
     *
     * @param x
     * @return pi(x)
     */
    public static int smallPi(long x){
        int pos = Arrays.binarySearch(primes,x);
        if(pos<0) return -pos-1;
        else return pos+1;
    }

    /**
     * La función eratosthenesInterval() usa el algoritmo de criba
     * parcial para devolver los números en el intervalo [m,n] cribados
     * con los primos en la tabla prime menores que length.
     * Notar que los valores devueltos no son necesariamente primos.
     *
     * @param m Extremo inferior del intervalo
     * @param n Extremo superior del inervalo
     * @param prime Array con primos
     * @param length Índice hasta el que se consulta el array prime
     * @return Array de números en [m,n] cribados con los primeros
     *         length primos
     */
    public static long[] eratosthenesInterval(long m, long n, long[] prime,
         int length){

        m+=m%2==0?1:0;
        n-=n%2==0?1:0;

        int stop = (int) (n-m+2)/2;
        BitSet isMult = new BitSet(stop);

        int i=1;
        long p=prime[i];
        long p2=p*p;
        long start;
        do{
            if(p2<m){
                long q=2*p;
                start=(m/q)*q+p;
                if (start<m) start+=q;
            } else start=p2;

            int j=(int)(start-m)/2+1;
```

```
            while(j<=stop){
                isMult.set(j-1);
                j+=p;
            }

            p=prime[++i]; p2=p*p;
        }
        while(i<length && p2<=n);

        int primesCount=0;
        long[] newPrimes =new long[stop];
        for(int j=0; j<stop; j++){
            if (isMult.get(j)==false) newPrimes[primesCount++]=(2*j+m);
        }

        return Arrays.copyOf(newPrimes, primesCount);
    }

    /* Esta función es una versión sobrecargada de la función anterior.
     * En vez de devolver un array, almacena en array[] los valores
     * calculados empezando a guardarlos en el índice arrayStart. */
    public static void eratosthenesInterval(long[] array, int arrayStart,
        long m, long n, long[] prime, int length){

        m+=m%2==0?1:0;
        n-=n%2==0?1:0;

        int stop = (int) (n-m+2)/2;
        BitSet isMult = new BitSet(stop);

        int i=1;
        long p=prime[i];
        long p2=p*p;
        long start;
        do{
            if(p2<m){
                long q=2*p;
                start=(m/q)*q+p;
                if (start<m) start+=q;
            } else start=p2;

            int j=(int)(start-m)/2+1;
            while(j<=stop){
                isMult.set(j-1);
                j+=p;
            }

            p=prime[++i]; p2=p*p;
        }
        while(i<length && p2<=n);

        int primesCount=arrayStart;
        for(int j=0; j<stop; j++){
            if (isMult.get(j)==false){
                array[primesCount++]=(2*j+m);
            }
        }
    }
```

```

    /**
     * Esta función criba los enteros en el intervalo [1,n]
     * almacenando en el array array[] los primos en ese intervalo.
     *
     * @param array Array donde se almacenan los primos.
     * @param n Extremos superior hasta el que cribar.
     */
    public static void eratosthenes(long[] array, int n){
        int stop=(n-1)/2;

        BitSet isMult = new BitSet(stop+1);

        int p=3, p2=9;
        int k=1;
        do{
            for(int i=(p2-1)/2; i<=stop; i+=p){
                isMult.set(i);
            }

            do k++; while(isMult.get(k) && k<stop);

            p=2*k+1;
            p2=p*p;

        } while(p2<=n);

        int primesCount=1;
        array[0]=2;
        for(int i=1; i<=stop; i++){
            if (isMult.get(i)==false) array[primesCount++]=(2*i+1);
        }
    }

    /* Se usa el método de Herón para calcular la raíz cuadrada de un
     * número entero x. De forma análoga se calcula la raíz cúbica. */
    public static long intSqrt(long x){
        long x0=x/2; //Es mejor aprox. inicial 2^[log_2(n)/2+1]
        long x1=(x0+x/x0)/2;
        while(x1<x0){
            x0=x1;
            x1=(x0+x/x0)/2;
        }
        return x0;
    }

    public static long intCbrt(long x){
        long x0=2*x/3;
        long x1=(2*x0+x/(x0*x0))/3;
        while(x1<x0){
            x0=x1;
            x1=(2*x0+x/(x0*x0))/3;
        }
        return x0;
    }
}
```