Investigación y docencia

Algoritmos de aritmética elemental

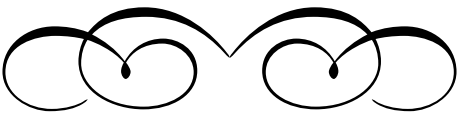
por N. Aguilera

Estas notas forman parte de la colección de apuntes en la sección *Investigación y Docencia* de la Olimpíada Matemática Argentina, disponibles vía internet en http://oma.org.ar/invydoc.

Fecha de esta versión: 18 de enero de 2019

Contenidos

Figuras



1. Introducción

Entre las muchas y lindas publicaciones de la OMA se encuentra una joya de Enzo Gentile: *Aritmética elemental en la formación matemática*.

Los temas abordados por ese libro han adquirido importancia práctica en años recientes debido a que gran parte de la seguridad de internet depende de algoritmos fundados en las propiedades de divisibilidad de enteros.

Basándonos en el libro de Gentile, en este apunte veremos algunos algoritmos computacionales sencillos, integrando teoría y práctica, incluyendo fracciones continuas y versiones básicas de algunas aplicaciones en criptografía.

Estas notas son una ampliación de los ejercicios que vimos en el Seminario Nacional de la OMA de 2018 realizado en La Falda, y están organizadas como ejercicios —una pobre imitación del estilo de Gentile— dejando para una sección final algunos resultados que requieren algo más de conocimientos. Entremezclados con los ejercicios aparecerán notas técnicas y comentarios históricos.

Suponemos que el lector está familiarizado con elementos básicos del libro de *Gentile*, especialmente congruencias, así como con rudimentos de programación.

Trabajaremos con el lenguaje Python, tratando de usar estructuras disponibles en la mayoría de los lenguajes de programación. En la *página de estas notas en la OMA*[(1)] hay enlaces a archivos de Python con versiones —nada profesionales— de algunos algoritmos que vemos acá.

1.1. ¿Por qué la computadora?

Además de la actualidad práctica de algunos temas que veremos, destaquemos dos razones válidas en todos los casos:

- El pensamiento algorítmico es parte del pensamiento matemático.

(1) http://oma.org.ar/invydoc/algoarit.html

Desde ya, la construcción del máximo divisor común de Euclides o la criba de Eratóstenes se encuentran entre los primeros algoritmos que recuerda la historia y aún se siguen usando.

Fuera del ámbito numérico, la geometría clásica nos habla de construcciones con regla y compás, que no son sino algoritmos para «fabricar» algún objeto.

Más aún, gran parte de las demostraciones de teoremas apelan a construcciones, más concretas o más abstractas. Por ejemplo, una de las demostraciones más bellas es la dada por Euclides de la existencia de infinitos primos: si $p_1, p_2, \ldots, p_n$ son primos, «fabricamos» el producto y le sumamos 1, resulta que debe haber otro primo pues ninguno de $p_1, p_2, \ldots, p_n$ divide a $p_1 \times \cdots \times p_n + 1$.

- La matemática experimental.

  La tecnología nos ha brindado herramientas cada vez más poderosas, haciéndonos olvidar muchas herramientas que usábamos no hace mucho. Pasó con la calculadora, que nos hizo olvidar de las tablas de logaritmos o las reglas de cálculo, y pasó con las aplicaciones de geometría dinámica, que superaron ampliamente la regla y el compás permitiéndonos ver cómo varían los objetos cuando se cambian algunos parámetros.

  El desarrollo y programación de algoritmos nos ayuda a entender más profundamente las ideas involucradas, hacer cálculos complicados o tediosos, y experimentar variando parámetros buscando patrones y realizando conjeturas.

En fin, desde el punto de vista del aprendizaje la computadora presenta un nivel intermedio entre la manipulación de objetos concretos y la abstracción, en cierto sentido similar al uso de la regla y el compás, y por otro lado siguiendo la moda de integrar conocimientos— conecta la matemática con la computación.

1.2. Bibliografía básica

El libro de *Gentile* será nuestra referencia básica para la teoría de números, especialmente para la primera parte del apunte. Niven, Zuckerman y Montgomery (1991), Rosen (1993) y Redmond (1996) presentan versiones más profundas de esa teoría, incluyendo algunos aspectos computacionales. El «clásico» para teoría de números es libro de Hardy

y Wright (2008), que se actualiza periódicamente. Crandall y Pomerance (2005) presentan un tratamiento computacional de divisibilidad, con vistas a las aplicaciones concretas de criptografía.

Un libro popular de programación, con muchos temas relacionados a matemáticas, es el de Cormen, Leiserson, Rivest y Stein (2009) (Rivest es la «R» del método RSA). La «biblia» de Knuth, *The art of computer programming*, abarca mucho más, y acá nos interesa principalmente el volumen 2 (Knuth, 1998).

Si bien ha quedado algo desactualizado porque trabaja con el lenguaje Pascal, el libro de Engel (1993) está pensado como introducción a la programación para profesores de secundaria. El clásico libro de Olds (1963) sobre fracciones continuas y ecuaciones diofánticas sencillas también está pensado para estudiantes y profesores de nivel medio.

1.3. Algunos comentarios históricos

Es un buen momento para señalar algunos comentarios históricos relacionados con lo que acabamos de expresar y destacar algunos nombres que aparecerán con frecuencia.

Acá nos detenemos un poco en la anécdota, el libro de *Gentile* tiene una exposición mucho más profunda.

- Las palabras *algoritmo* y *guarismo* derivan del nombre de Abu Ja'far Muhammad ibn Musa al-Khwarizmi', castellanizado como «al-Juarismi». Se presume que nació alrededor de 780 en Bagdad, cuando Harun al-Rashid —el califa de *Las mil y una noches*— comenzaba su califato, y murió en 850.

  Al-Juarismi escribió y tradujo varios textos de matemática y otras ciencias del griego al árabe. El más importante de ellos es *Hisab al-jabr w'al-muqabala*, de donde surge la palabra *álgebra* con la que ahora designamos a esa rama de la matemática.

  Alrededor de 820 escribió un tratado sobre números indoarábigos que se ha perdido (ni siquiera se conoce su título), pero en el siglo XII se hizo una traducción parcial al latín llamada *Algoritmi de numero Indorum*, para indicar su autor, dando lugar a la palabra *algoritmo*.

- Aunque hay discusión sobre si se trata de una persona o un grupo, Euclides de Alejandría (alrededor de 325–265 a. C.) escribió una

serie de libros de enorme influencia en las matemáticas, inclusive en las actuales. En *Los Elementos* presenta los principios de lo que llamamos geometría euclidiana a partir de un pequeño conjunto de axiomas. Estos axiomas perduraron hasta fines del siglo XIX (con las modificaciones de Pasch y la aparición de las geometrías no euclidianas), y en nuestro país los libros de texto de geometría en la escuela secundaria siguieron su presentación hasta bien avanzado el siglo XX (hasta la aparición de la «matemática moderna»).

En la época de Euclides los números representaban longitudes de segmentos, y de allí que *Los Elementos* tratara dentro de la geometría cuestiones que hoy consideraríamos como de teoría de números.

En el original griego, *Euclides* significa «renombrado» o «buena fama» (eu: *bien*, kleos: *fama*).

- Eratóstenes de Cirene (276 a. C.–194 a. C.) fue matemático, poeta, astrónomo y teórico de la música. Inventó la disciplina geografía, incluyendo terminología que usamos hoy, y su erudición lo llevó a ser el bibliotecario en jefe de la famosa Biblioteca de Alejandría.

  Fue el primero en medir con una buena aproximación la circunferencia de la Tierra, ¡y Colón usaba un huevo 1700 años después!

- Las ecuaciones algebraicas (polinómicas) con coeficientes enteros donde sólo interesan las soluciones enteras se llaman *diofánticas* (*Gentile* las llama *diofantinas*) en honor a Diofanto de Alejandría (aproximadamente 200–284) quien fue el primero en estudiar sistemáticamente estas ecuaciones, siendo autor del influyente libro *La Aritmética*. El libro originalmente fue escrito en griego, y recién tuvo amplia difusión en Europa cuando Claude Gaspard Bachet de Méziriac (1581–1638) la tradujo al latín publicándolo en 1621.

  Entre las ecuaciones diofánticas más sencillas está la ecuación lineal $ax + by = c$ a la que dedicaremos buena parte del apunte y que estudiamos con detalle en la sección 9.

  Pasando a las no lineales, posiblemente las que han suscitado mayor atención son las de la forma

$$x^n + y^n = z^n, \tag{1.1}$$

  donde n, x, y y z son naturales.

  Cuando $n = 2$, las soluciones de (1.1) forman una *terna pitagórica*

que exploramos en el ejercicio 7.12.

- Pierre de Fermat (1607–1665) será nombrado varias veces en estos apuntes.

  Tenía como costumbre enunciar resultados sin demostración en la correspondencia con sus colegas a modo de desafío, muchos de los cuales fueron luego demostrados rigurosamente por matemáticos de la talla de Euler. Aunque aquí veremos algunas de sus contribuciones en teoría de números, también fue el predecesor del cálculo infinitesimal, poco antes de Newton y Leibniz (lo que trajo disputas con Descartes y Wallis), y además hizo aportes en física como el principio de propagación de la luz que lleva su nombre.

  También escribió varios de sus resultados en los márgenes de su copia de la traducción de Bachet de *La Aritmética* de Diofanto, a modo de comentarios. Como es sabido, en 1637 en una de esas notas manifestó que para $n > 2$ la ecuación (1.1) no tiene soluciones, expresando

  > *encontré una demostración maravillosa, pero el margen del libro es demasiado pequeño para contenerla.*

  Se sospecha que posiblemente él mismo haya encontrado alguna falla en su razonamiento pues no volvió a mencionar el resultado. Es más, dejó una demostración de la imposibilidad para el caso $n = 4$, haciendo un uso profundo del *método de descenso* ya conocido por Euclides, técnica que muchos matemáticos emplearon posteriormente. Varios casos particulares de la imposibilidad de soluciones de la ecuación (1.1) fueron demostrados, hasta que en 1994 R. Taylor y A. Wiles demostraron la imposibilidad de soluciones para $n > 2$, ¡más de 350 años después del enunciado de Fermat!

- Otro que nombraremos muchas veces es Leonhard Euler (1707–1783), uno de los más grandes y prolíficos matemáticos de todos los tiempos.

  Euler hizo contribuciones en todas las áreas de las matemáticas de su época e inició otras nuevas, como la teoría de grafos y la topología al resolver en 1736 el famoso problema de los siete puentes de Königsberg.

  Fue tan grande su influencia que hoy usamos notaciones que el

ideó, como la de π ($= 3.14159\ldots$) (del griego *perypheria* o circunferencia), i ($= \sqrt{-1}$) (por imaginario), y e ($= 2.71828\ldots$) (del alemán *einhalt* o unidad).

- Según se dice, Johann Carl Friederich Gauss (1777–1855) tenía 8 años cuando el maestro de la escuela le dio como tarea sumar los números de 1 a 100 para mantenerlo ocupado (y que no molestara). Sin embargo, hizo la suma muy rápidamente al observar que la suma era 50×101.

  Las contribuciones de Gauss van mucho más allá de esta anécdota, quizás apócrifa. Sus trabajos son tan profundos y en tantas ramas de las matemáticas que le valieron el apodo de «príncipe de los matemáticos». Para algunos, fue el más grande matemático de todos los tiempos.

  En 1798 (a los 21 años) escribió el extremadamente influyente libro *Disquisitiones Arithmeticae* (publicado en 1801), en el que empieza por introducir la noción de congruencia con la notación $\equiv$ que usamos, y demuestra muchos resultados profundos en teoría de números. Nuestro teorema 13.15 y el mismo nombre *raíz primitiva*, que atribuye a Euler, están bien al principio del libro.

  No sólo fue decisiva su contribución en matemáticas, también lo fue en astronomía y física (el «gauss» es una medida de magnetismo).

1.4. Notaciones y otras cuestiones previas

Ponemos aquí convenciones y notaciones que usamos, algunas diferentes a las usadas en el libro de *Gentile*.

- $\mathbb{Z}$ es el conjunto de enteros, $\mathbb{Z} = \{0, 1, -1, 2, -2, \ldots\}$ y $\mathbb{N} = \{1, 2, 3, \ldots\}$. es el conjunto de números naturales formado por los enteros positivos. $\mathbb{R}$ es el conjunto de números reales.
- Flechas. $\Rightarrow$: implica (o sólo si), $\Leftrightarrow$: si y sólo si, $\Leftarrow$: si.
- Los números primos son naturales, y en particular mayores que 1. En ocasiones, *Gentile* considera que $-2, -3, -5, \ldots$ también son primos.
- Para $x \in \mathbb{R}$, usamos la función *piso*,

$$\lfloor x \rfloor = \max\{z \in \mathbb{Z} : z \leq x\},$$

y la función *techo*,

$$\lceil x \rceil = \min\{z \in \mathbb{Z} : z \geq x\},$$

y desde ya que $\lfloor x \rfloor = \lceil x \rceil$ si y sólo si $x \in \mathbb{Z}$.

Por supuesto, la función *parte entera* $[x]$ que usa *Gentile* coincide con la función piso.

Las nomenclaturas *piso* y *techo* son más comunes en matemática discreta y computación. Además, acá usamos corchetes para fracciones continuas.

- Si $a \in \mathbb{Z}$ y $b \in \mathbb{N}$, a veces indicaremos por $\text{coc}(a,b)$ al cociente y por $\text{resto}(a,b)$ el resto de la división *entera* de a por b, $0 \leq \text{resto}(a,b) < b$. Equivalentemente,

$$\text{coc}(a,b) = \left\lfloor \frac{a}{b} \right\rfloor, \tag{1.2}$$

de modo que $a = \text{coc}(a,b) \times b + \text{resto}(a,b)$.

En Python, la división común es `a/b`, la división entera $\text{coc}(a,b)$ es `a // b` y $\text{resto}(a,b)$ es `a % b`. Es usual en computación llamar *módulo* a la operación que nosotros llamamos *resto*.

- Para $a,b \in \mathbb{Z}$, $a \mid b$ significa que existe $c \in \mathbb{Z}$ tal que $b = a \times c$. Decimos en este caso que a divide a b o b es múltiplo de a o $\text{resto}(b,a) = 0$. La negación es $a \nmid b$.
- Si n es un entero mayor que 1 y $a,b \in \mathbb{Z}$, $a \equiv b \pmod{n}$ o $a \underset{n}{\equiv} b$ indica que $\text{resto}(a,n) = \text{resto}(b,n)$, o equivalentemente, $n \mid (a-b)$.
- A diferencia de *Gentile*, usamos la notación $\text{mcd}(a,b)$ en vez de $(a;b)$ para el máximo común divisor. Análogamente, para el mínimo común múltiplo usamos $\text{mcm}(a,b)$ en vez de $[a;b]$.
- Usamos punto y no coma decimal para no confundirnos al trabajar con la computadora: 123.456 tiene parte entera 123. Para separar miles en el texto (pero no en la computadora) a veces usamos un espacio pequeño: 123 456 (sin espacios sería 123456) es un entero con 6 cifras decimales.
- `Con este tipo de letra` indicaremos instrucciones de Python, y no haremos diferencia entre una variable `n` en Python y su versión matemática n.

- Para no apartarnos demasiado de otros lenguajes de programación, no usaremos *todas* las posibilidades de Python. Por ejemplo, evitaremos diccionarios y conjuntos.
- $\log_b x$ es el logaritmo en base b de x:

$$y = \log_b x \quad \Leftrightarrow \quad b^y = x.$$

Si no se indica la base es porque estamos considerando el *logaritmo natural* de base $e = 2.71828\ldots$, $\log x = \log_e x$ (algunos autores usan ln).

La definición del número irracional e involucra temas de análisis matemático (límites o integrales). Por ejemplo, con límites podemos definir

$$e \stackrel{\text{def}}{=} \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n.$$

En matemáticas y en economía (de donde proviene el límite anterior) es común usar la base e, en la vida diaria tal vez es más común la base 10, y en computación se usa la base 2, representando (casi) la cantidad de bits necesarios para representar un número entero no negativo.

En Python, valores aproximados de e y π y las funciones trascendentales como logaritmos o trigonométricas están en el módulo `math`:

```
import math
print("e:", math.e)
print("pi:", math.pi)
print("logaritmo natural de 10:", math.log(10))
print("logaritmo en base 10 de 2:", math.log10(2))
print("logaritmo en base 2 de 10:", math.log2(10))
print("2 a la (log2 10):", pow(2, math.log2(10)))
print("seno de 30 grados:",
      math.sin(30 * math.pi / 180))
```

2. Repasando cosas básicas

E 2.1 (buena ordenación de $\mathbb{N}$). Una propiedad importante de los números naturales que resulta muy útil desde la teoría es el *principio*

de buena ordenación:

1. *Todo subconjunto no vacío de* $\mathbb{N}$ *tiene un mínimo.*

Equivalentemente,

2. *Todo subconjunto no vacío de* $\mathbb{N}$ *que está acotado superiormente tiene un máximo.*

> ✍ Decimos que $A \subset \mathbb{R}$, $A \neq \emptyset$, está *acotado superiormente* si existe $M \in \mathbb{R}$ tal que $a \leq M$ para todo $a \in A$.

¿Podrías demostrar la equivalencia entre los enunciados en 1 y 2? ✂

E 2.2. Un corolario importante del principio de buena ordenación de $\mathbb{N}$ es el *principio de inducción*: si $A \subset \mathbb{N}$ es tal que

- $1 \in A$,
- si $n \in A$ entonces $n+1 \in A$,

entonces $A = \mathbb{N}$.

Demostrar el principio de inducción a partir del principio de buena ordenación.

> ✍ Que el principio de inducción sea consecuencia del principio de buena ordenación depende de cómo se defina $\mathbb{N}$.[(2)] Por ejemplo, en la aritmética de Peano buena ordenación es una consecuencia de inducción.

✂

E 2.3. Sean A y B conjuntos no vacíos, $f : A \to B$ biyectiva. Entonces existe $g : B \to A$ tal que $g \circ f : A \to A$ es la identidad en A (o sea, $g \circ f(x) = x$ para todo $x \in A$), y $f \circ g : B \to B$ es la identidad en B.

g es la *función inversa* de f, y normalmente se la indica por f^{-1}. ✂

E 2.4. Sean A, B y C conjuntos no vacíos, $f : A \to B$, $g : B \to C$, $h : A \to C$ la composición de f y g, $h = g \circ f$.

a) Si f y g son inyectivas, h también lo es.
¿Podrían ser f y h inyectivas pero no g?

b) Si f y g son suryectivas, h también lo es.

c) Si f y g son biyectivas, entonces h también lo es.
¿Podrian ser g y h biyectivas pero no f? ✂

(2) O sea, es una cuestión de principios.

Indicaremos por $\mathbb{I}_n$ al conjunto de los primeros n naturales,

$$\mathbb{I}_n = \{j \in \mathbb{N} : j \leq n\} = \{1, 2, \ldots, n\}.$$

Un conjunto no vacío A es *finito* si existe $f : \mathbb{I}_n \to A$ biyectiva, en cuyo caso decimos que el *cardinal* de A es n, y lo indicamos mediante

$$\#(A) = n.$$

También decimos que el conjunto vacío es finito y su cardinal es 0, $\#(\emptyset) = 0$.

✍ La notación $|A|$ para el cardinal de A es bastante común en matemática discreta.

E 2.5. Sean A y B conjuntos finitos no vacíos. Entonces $\#(A) = \#(B)$ si y sólo si existe una biyección entre A y B. ✂

E 2.6 (cardinal de la unión disjunta). La siguiente es una propiedad fundamental de «conteo»:

Si $A_1, A_2, \ldots, A_k$ es una familia de conjuntos finitos disjuntos, entonces

$$\#\left(\bigcup_{i=1}^{k} A_i\right) = \sum_{i=1}^{k} \#(A_i)$$

o en palabras, el cardinal de la unión es la suma de los cardinales.

¿Podrías demostrar esta propiedad usando la definición de conjunto finito y su cardinal? ✂

Otro resultado de «conteo» que usaremos es el cardinal del producto cartesiano. Recordemos que si A y B son conjuntos no vacíos, entonces su producto cartesiano es

$$A \times B = \{(a, b) : a \in A, b \in B\}.$$

E 2.7 (cardinal de producto). Demostrar que si A y B son conjuntos no vacíos,

$$\#(A \times B) = \#(A) \times \#(B).$$

Aclaración: poniendo $m = \#(A)$ y $n = \#(B)$, se pretende encontrar una biyección entre $\mathbb{I}_{m \times n}$ y $A \times B$. ✂

E 2.8 (Principio del casillero). Si n objetos deben repartirse en m casilleros, $m \le n$, entonces hay un casillero que contiene al menos $\lceil n/m \rceil$ objetos.

Sugerencia: considerar a los casilleros como conjuntos y usar el ejercicio 2.6.

✍ La versión más conocida es cuando $n = m + 1$, en cuyo caso hay al menos un casillero que tiene 2 o más elementos. ✂

E 2.9. Sean A y B conjuntos finitos y $f : A \to B$.

a) Si f es inyectiva, entonces $\#(A) \le \#(B)$.

b) Si f es suryectiva, entonces $\#(A) \ge \#(B)$.

c) Si $\#(A) = \#(B)$ entonces son equivalentes:

i) f es inyectiva.

ii) f es suryectiva.

iii) f es biyectiva.

✍ Comparar con el ejercicio 2.5. ✂

Para terminar el repaso, conviene recordar que las congruencias son relaciones de equivalencia compatibles con las operaciones aritméticas.

E 2.10 (*Gentile*). *a*) $a \underset{n}{\equiv} a$ (reflexiva).

b) $a \underset{n}{\equiv} b \Rightarrow b \underset{n}{\equiv} a$ (simétrica).

c) $a \underset{n}{\equiv} b$ y $b \underset{n}{\equiv} c \Rightarrow a \underset{n}{\equiv} c$ (transitiva),

d) $a \underset{n}{\equiv} b$ y $c \underset{n}{\equiv} d \Rightarrow a + c \underset{n}{\equiv} b + d$, $a \times c \underset{n}{\equiv} b \times d$. ✂

3. Sacando las cosquillas a Python

E 3.1 (*Gentile*). Sean

$$a = \sqrt[3]{2 + \frac{10}{9}\sqrt{3}}, \quad b = \sqrt[3]{2 - \frac{10}{9}\sqrt{3}}, \quad c = a + b.$$

a) Demostrar que a y b son irracionales.

b) Calcular c con Python o alguna calculadora común (que pueda calcular raíces)?

c) ¿Es c irracional?

✍ Internamente, la máquina trabaja con números racionales (y de la forma $m/2^n$ con m y n enteros) y sólo puede tener una aproximación de los valores de los irracionales a y b. Es más, aún cuando a y b fueran racionales y representables, los cálculos de raíces cuadradas y cúbicas dan sólo aproximaciones a los valores verdaderos.

Por lo tanto, es imposible que la máquina dé el valor correcto de $c = a+b$, salvo que haya algún redondeo en alguna parte o un milagro haga que los errores numéricos se compensen.

Curiosamente, en algunas computadoras Python da la respuesta correcta para c desde la teoría: 2.

Para profundizar el tema, podemos tratar cada sumando por separado poniendo:

```
a = (2 + 10 * 3**(1/2) / 9)**(1/3)
b = (2 - 10 * 3**(1/2) / 9)**(1/3)
c = a + b
print("a:", a)
print("b:", b)
print("c = a + b:", c)
print("c como cociente:", c.as_integer_ratio())
```

donde la última instrucción escribe numerador y denominador de la representación interna como fracción.

En mi máquina los resultados son:

```
a: 1.5773502691896257
b: 0.4226497308103747
c = a + b: 2.0000000000000004
c como cociente: (4503599627370497, 2251799813685248)
```

Observemos que $2251799813685248 = 2^{51}$: la máquina sólo puede representar fracciones donde el denominador es una potencia de 2 (y sólo algunas pocas). ✂

E 3.2 (años bisiestos). Desarrollar una función para decidir si un año dado es o no bisiesto, imprimiendo el resultado.

✍ Según el calendario gregoriano que usamos, los años bisiestos son aquellos divisibles por 4 excepto si divisibles por 100 pero no por 400. Así, el año 1900 no es bisiesto pero sí lo son 2012 y 2000.

Este criterio fue establecido por el Papa Gregorio XIII en 1582, pero en el ejercicio adoptamos este criterio para cualquier año, anterior o

posterior a 1582.

✍ Julio César (101 o 100–44 a. C.) cambió el calendario egipcio, que estaba basado en un año de exactamente 365 días, a un nuevo calendario, el *juliano*, con años bisiestos cada 4 años para reflejar mejor la verdadera longitud del año.

Sin embargo, cálculos posteriores mostraron que la longitud del año es aproximadamente 365.2422 días.

Con el paso de los siglos, las diferencias anuales de 0.0078 días en promedio se fueron acumulando, y hacia el año 1582 el Papa Gregorio comenzó el calendario que usamos ahora, con el cual el año promedio tiene 365.2425 días.

Curiosamente, se agregaron 10 días a la fecha, de modo que el 5 de octubre de 1582 pasó a ser el 15 de octubre (y nunca existieron los días entre el 6 y el 14 de octubre de ese año).

No todos los países del mundo adoptaron este calendario inmediatamente. En Gran Bretaña y lo que es ahora Estados Unidos de Norteamérica, se adoptó recién en 1752, por lo que se debieron agregar 11 días más; Japón cambió en 1873, Rusia en 1917 y Grecia en 1923.

✍ A veces con un pequeño esfuerzo podemos hacer el cálculo más eficiente. Un esquema para resolver el problema anterior, si *anio* es el año ingresado, es

```
if anio % 400 == 0:
    print(anio, 'es bisiesto')
elif anio % 100 == 0:
    print(anio, 'no es bisiesto')
elif anio % 4 == 0:
    print(anio, 'es bisiesto')
else:
    print(anio, 'no es bisiesto')
```

Sin embargo, el esquema

```
if anio % 4 != 0:
    print(anio, 'no es bisiesto')
elif anio % 100 != 0:
    print(anio, 'es bisiesto')
elif anio % 400 != 0:
    print(anio, 'no es bisiesto')
else:
    print(anio, 'es bisiesto')
```

es más eficiente, pues siendo que la mayoría de los números no son múltiplos de 4, en la mayoría de los casos haremos sólo una pregunta con el segundo esquema pero tres con el primero. ✂

E 3.3. Construir una función `edad(nac, hoy)` para imprimir la edad de una persona (en años) dadas las fechas de nacimiento y la actual, donde cada argumento es de la forma `[a, m, d]` por año, mes y día, tanto para la fecha de nacimiento como para la fecha actual.

✍ Una función de Python termina en cuanto se ejecuta una instrucción `return`, por lo que en este ejercicio podría usarse `if` sin `else` ni `elif`. ✂

E 3.4 (*Gentile*). Hacer una función para determinar el día de la semana (domingo, lunes,..., sábado) de una fecha ingresada en la forma `[a, m, d]` (año, mes, día), y aplicarla para encontrar los días correspondientes al 18 de septiembre de 1808, 9 de julio de 1816, 6 de junio de 1944 (el «día D» de la segunda guerra mundial). ✂

E 3.5. El *factorial* de $n \in \mathbb{N}$ se define como

$$n! = 1 \times 2 \times \cdots \times n,$$

y por conveniencia ponemos $0! = 1$.

Una versión inductiva de la definición podría ser:

$$0! = 1, \quad n! = n \times (n-1)! \text{ si } n \in \mathbb{N}.$$

a) Definir una función `factorial(n)` para calcular $n!$ (donde $n \in \mathbb{Z}$, $n \geq 0$) usando un lazo `for`.

¿Cuántos asignaciones y multiplicaciones se realizan?

b) Definir una función para calcular el coeficiente binomial

$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

donde $k \in \mathbb{Z}$, $0 \leq k \leq n$, haciendo no más de $2(k-1)$ multiplicaciones y una división cuando $k \leq n/2$, y reduciendo el caso $k > n/2$ al anterior: ¡no usar el cociente de los factoriales!

c) La *fórmula de Stirling* dice que para n grande

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \tag{3.1}$$

donde $e = 2.71828\ldots$ es la base de los logaritmos naturales.

Verificar la bondad de esta aproximación para $n = 10$, 20, 30, 40 y 50 tomando el cociente entre $n!$ y la aproximación.

✍ Según la Encyclopaedia Britannica, J. Stirling publicó la fórmula en 1730. Algunos, como Wikipedia, sostienen que A. de Moivre había encontrado previamente el comportamiento asintótico sin determinar la constante $\left(\sqrt{2\pi}\right)$. Sin embargo, la publicación de de Moivre es de 1733. ✂

E 3.6 (complejidad del factorial). En este ejercicio raspamos la superficie del problema de complejidad de algoritmos.

En general interesa saber la cantidad de pasos (asignaciones, comparaciones u operaciones aritméticas) en términos de los tamaños de entrada y salida, eventualmente considerando también el espacio de memoria (en cantidad de bits) usada.

Surge la duda de si la operación $(a + b) \times c$ se realiza en uno o dos pasos (la suma y después el producto). Acá no nos interesará mucho hacer estas distinciones, y básicamente pensamos que un paso equivale a un renglón del programa.

Así, hablamos del *orden* del algoritmo, indicado por O, y simplemente consideramos el término de mayor orden, eliminando constantes. Por ejemplo, si la cantidad de pasos es $5n^2$ o $3n^2 - 5n + \log n$, simplemente diremos que el algoritmo es $O(n^2)$.

✍ Si bien acá consideramos que cada asignación, comparación u operación aritmética equivale a un paso, a medida que los tamaños aumentan sumas y multiplicaciones llevan más «tiempo». Por ejemplo, para multiplicar dos números con unos pocos miles de dígitos, digamos m dígitos cada uno, se usan algoritmos que tardan del orden de $m^{1.5}$ (aproximadamente). Para más cantidad de dígitos se consideran algoritmos más eficientes y también más complicados.

Para simplificar, acá pensaremos que las operaciones, asignaciones y comparaciones se realizan todas en «un paso».

✍ En muchos casos, la cantidad de pasos a realizar depende de otros factores, no sólo del tamaño de la entrada.

Por ejemplo, la cantidad de pasos para encontrar el máximo común divisor entre 2^n y 2^{n+1} ($n \in \mathbb{N}$) no depende esencialmente de n.

Acá adoptamos el *criterio del peor caso*, encontrando una cota superior en términos de los tamaños involucrados.

a) Ver que la cantidad de bits para representar el natural n es $1 + \lfloor \log_2 n \rfloor$.

Decimos entonces que el tamaño de n es $O(\log n)$, donde no importa cuál base usamos pues los logaritmos difieren sólo en una constante: $\log_a x = (\log_a b)\,(\log_b x)$.

b) Dar una estimación del tamaño (en bits) de $n!$ usando la fórmula de Stirling (3.1).

c) Volviendo a la función `factorial` definida en el ejercicio 3.5, ¿cómo se comparan el tamaño (cantidad de bits) de la entrada, el de la salida y el número de pasos realizados?, ¿podríamos decir que la cantidad de pasos es exponencial en el tamaño de la entrada?, si no, ¿cómo se justifica el tamaño de la salida?

d) En el análisis de complejidad, también hay que tener en cuenta la cantidad de memoria usada. A veces se usan pocos pasos y mucha memoria, y para ahorrar memoria se cambia el algoritmo usando muchos más pasos.

En el caso de la función `factorial`, ¿qué cantidad de bits de memoria se necesitan?, ¿es comparable al tamaño de la entrada o salida?

E 3.7 (números de Fibonacci). Los números de Fibonacci f_n se definen mediante:

$$f_1 = 1, \quad f_2 = 1, \quad \text{y} \quad f_n = f_{n-1} + f_{n-2} \quad \text{para } n \geq 3,$$

obteniendo la sucesión $1, 1, 2, 3, 5, 8, \ldots$

a) Construir una función `fibo(n)` para encontrar el n-ésimo número de Fibonacci usando un lazo y sumas de dos consecutivos.

b) De la teoría general de relaciones de recurrencia, se ve que vale la fórmula de Euler-Binet:

$$f_n = \frac{\tau^n - (\tau')^n}{\sqrt{5}} = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n \sqrt{5}}, \tag{3.2}$$

donde $\tau = (1+\sqrt{5})/2$ es el *número de oro*, $\tau' = (1-\sqrt{5})/2 = -1/\tau$.

✍ Usamos la notación τ en vez de ϕ para no confundir con la función de Euler que veremos más tarde.

Demostrar esta identidad usando inducción.
¿Podrías demostrarla directamente (sin usar inducción)?

c) Calcular los números de Fibonacci usando la fórmula (3.2), definiendo una función `binet(n)`.

d) Comparar el número de Fibonacci f_n con el redondeo (`round`) de

$$\frac{\tau^n}{\sqrt{5}} = \frac{\left(1+\sqrt{5}\right)^n}{2^n\,\sqrt{5}}.$$

¿A partir de qué n los valores coinciden?, ¿podrías justificarlo?

✍ Leonardo Bigollo es el verdadero nombre de Leonardo de Pisa (1180–1250), posteriormente conocido como Fibonacci (contracción de las palabras «hijo de Bonacci»).

Fibonacci escribió el libro *Liber Abaci* en 1202, pero no se conservan copias de ese original. En 1227 se hizo una segunda versión en la que aparece como contribución algo menor el problema de los conejos y los números que posteriormente se nombrarían en su honor. Ciertamente, la mayor contribución del libro fue la introducción del sistema de numeración indo-arábigo en Europa (en el libro aparecen comentarios sobre los trabajos de al-Juarismi que mencionamos en la Introducción).

Fibonacci publicó varios tratados sobre teoría de números, geometría y la relación entre ellos.

✍ Jacques Binet (1786–1856) publicó la fórmula para los números de Fibonacci en 1843, pero ya había sido publicada por Euler en 1765, y de ahí el nombre de la relación. Sin embargo, A. de Moivre ya la había publicado (y en forma más general) en 1730.

✍ Mucho después de Fibonacci, se observó que los números f_n aparecen en muy diversos contextos, algunos insospechados como en la forma de las flores del girasol, y son de importancia tanto en las aplicaciones prácticas como teóricas.

Por ejemplo, han sido usados para resolver problemas de confiabilidad de comunicaciones y en estructuras de datos en computación.

Más adelante veremos algunas aplicaciones teóricas sencillas, relacionadas con la eficiencia del algoritmo de Euclides y las fracciones continuas.

Acá vale la pena mencionar que usando la tasa de crecimiento de los números de Fibonacci, Y. Matijasevich demostró en 1970 (cuando tenía 22 años) que no existe un algoritmo que pueda determinar si una ecuación diofántica polinomial arbitraria tiene soluciones enteras. Este es el décimo en la lista de problemas presentados por D. Hilbert en el Congreso Internacional de Matemáticas de 1900, lista que moldeó gran parte de la matemática del siglo XX. ✂

4. Divisibilidad

E 4.1. El *algoritmo de la división* entre los enteros positivos a y b dice que existen únicos enteros q y r tales que

$$a = qb + r \quad \text{y} \quad 0 \le r < b.$$

a) Justificar este «algoritmo» definiendo

$$\mathcal{Q} = \{u \in \mathbb{Z} : u \ge 0 \text{ y } u \times b \le a\},$$

viendo que $\mathcal{Q}$ es no vacío y tiene máximo, y

$$q = \text{máx}\, \mathcal{Q}.$$

Desde ya que r queda determinado una vez conocido q.

b) ¿Podría plantearse algo similar para obtener r como mínimo de algún conjunto (para después obtener q)?

✍ Euclides no enuncia el algoritmo de la división en *Los Elementos*, siempre hace las divisiones por restas sucesivas. ✂

Como ya mencionamos en la sección 1.4 $q = \text{coc}(a, b)$ es el resultado de la *división entera* de a por b.

E 4.2. En este ejercicio queremos definir una función para encontrar el cociente y resto de la división entre enteros positivos tomando restas sucesivas, siguiendo un esquema como:

```
q = 0
r = a
while r >= b:
    r = r - b   # lo que queda: ¡el resto!
    q = q + 1   # cantidad de restas
return q, r
```

a) Observar que siempre se mantiene el *invariante* $a = q \times b + r$: en cada paso r disminuye en b y q aumenta en 1.

¿Siempre termina el algoritmo?, ¿por qué cuando termina resulta $0 \le r < b$?, ¿qué pasa si $a < 0$ (y $b > 1$)?

b) Definir una función `qr(a, b)` usando estas ideas (retornando el par (q,r)), y verificar su comportamiento tomando distintos valores de a y b.

c) Comparar el comportamiento de `qr` con las funciones `/`, `//` y `divmod` de Python.

d) ¿Qué debería dar la función `qr` si se aplica a decimales? Verificarlo con distintas entradas.

E 4.3 (cifras). Supongamos que n es un entero positivo, y b es un entero, $b > 1$.

a) ¿Cómo se puede calcular la cantidad de cifras de n en base b?

b) ¿Cuántas cifras debería tener 0?

✍ Una posibilidad para *a*) (que no incluye a 0) es:

```
c = 0
while n > 0:
    n = n // b
    c = c + 1
# acá c es el número de cifras
```

Una variante para incluir 0 es un esquema «repetir hasta que»:

```
c = 0
while True:     # repetir
    n = n // b
    c = c + 1
    if n == 0:  # hasta que
        break
# acá c es el número de cifras
```

c) Construir una función para dar una lista de las cifras de un entero no negativo en base $b > 1$ lista (las cifras más significativas primero).

Verificar la cantidad de cifras usando `len`.

✍ Posiblemente sea más sencillo construir una lista y luego darla vuelta.

Recordemos que si `a` es una lista de Python, `a[:]` es una copia de la lista, `a[-1]` es el último elemento (si no vacía), `a[::-1]` es la lista «dada vuelta», y `a[-2::-1]` es la lista dada vuelta habiendo quitado primero el último elemento de `a`. Por otro lado `a.reverse()` da vuelta la lista *pero la modifica*.

d) Construir una función para que dadas las cifras de n en base b reconstruya el número n.

✍ La *regla de Horner* evalúa el polinomio

$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

en la forma

$$((\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots)x + a_1)x + a_0,$$

justamente la inversa del cálculo de los coeficientes que naturalmente se hace en el orden a_0 (primer resto), a_1 (segundo), etc.

e) Construir una función para «pasar» de una base a otra: se da una lista de las cifras de algún número natural en una base y se quiere obtener la lista de cifras en otra base.

Por ejemplo, la lista `[1, 3]` representa al número 13 en base 10, y en base 2 la lista de coeficientes es `[1, 1, 0, 1]`. ✂

E 4.4 (potencia binaria). Dados $x \in \mathbb{R}$ y $n \in \mathbb{N}$ queremos calcular x^n.

a) ¿Cuántas multiplicaciones se realizan con el siguiente esquema en Python?

```
p = 1
for i in range(n):  # hacer n veces:
    p = p * x       # multiplicar por x
return p
```

Observemos que si $n = 2^k$ ($k \in \mathbb{N}$) podemos escribir

$$x^n = x^{2^k} = \left(\ldots\left(\left(x^2\right)^2\right)^2\ldots\right)^2,$$

donde en la expresión a la derecha se eleva al cuadrado k veces. Contando cada cuadrado como una multiplicación, se realizan unas $k = \log_2 n$ multiplicaciones.

Aprovechamos esta idea escribiendo n en base 2,

$$n = \sum_{i=0}^{k} a_i 2^i,$$

y expresando

$$x^n = x^{a_0+2a_1+2^2a_2+\cdots+2^k a_k} = x^{a_0} \cdot \left(x^2\right)^{a_1} \cdot \left(x^4\right)^{a_2} \cdots \left(x^{2^k}\right)^{a_k}.$$

b) Si los coeficientes se conocen y están guardados en la lista a, un esquema posible es

```
if a[0] == 1:
    producto = x
else:
    producto = 1
potencia = x
for aj in a[1:]:    # aj = a[j]
    potencia = potencia * potencia    # = x**(2**j)
    if aj == 1:
        producto = producto * potencia
return producto
```

Hacer una función con estas ideas (con argumentos el entero no negativo x y la lista a).

c) Modificar la función anterior de modo que los argumentos sean x y n y la función va encontrando los coeficientes.

d) Comparando con el esquema inicial en *a*), ¿cuántas operaciones aritméticas (productos y divisiones) se realizan en cada caso?

✍ En Python, `pow(a, b)` (sin el tercer argumento) es simplemente a^b.

✍ Muchos lenguajes de programación, entre ellos Python, toman el resultado de 0^0 como 1 por convención.

E 4.5 (juego de Nim). En este ejercicio estudiamos el siguiente juego:

Dos jugadores, A y B, colocan un número arbitrario de fósforos sobre una mesa, separados en filas o grupos. El número

de filas y el número de fósforos en cada fila también son arbitrarios. El primer jugador, A, toma cualquier número de fósforos de un fila, pudiendo tomar uno, dos o hasta toda la fila, pero sólo debe modificar una fila. El jugador B juega de manera similar con los fósforos que quedan, y los jugadores van alternándose en sus jugadas. Gana el jugador que saca el último fósforo.

Veremos que, dependiendo de la posición inicial, uno u otro jugador tiene siempre una estrategia ganadora. Digamos que una disposición de los fósforos es *ganadora* para el jugador X, si *dejando* X los fósforos en esa posición, entonces no importa cuál sea la jugada del oponente, X puede jugar de forma tal de ganar el juego.

a) Ver que la posición en la cual hay dos filas con dos fósforos cada una, es ganadora.

b) Ver que la posición en donde hay 3 filas con 1, 2 y 3 fósforos respectivamente, es ganadora.

Para encontrar una estrategia ganadora, formamos una tabla expresando el número de fósforos de cada fila en binario, uno bajo el otro, poniendo en una fila final P si la suma total de la columna correspondiente es par, e I en otro caso. Por ejemplo, en las posiciones anteriores haríamos:

$$\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ \hline P & P \end{array} \qquad \text{y} \qquad \begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ \hline P & P \end{array}$$

En el último ejemplo, es conveniente poner 01 en vez de sólo 1 en la primera fila a fin de tener todas las filas con igual longitud.

Si la suma de cada columna es par decimos que la posición es *correcta*, y en cualquier otro caso decimos que la posición es *incorrecta*. Por ejemplo, la posición donde hay 1, 3 y 4 fósforos es incorrecta:

$$\begin{array}{lcc} | & \to & 0\ 0\ 1 \\ | \ | \ | & \to & 0\ 1\ 1 \\ | \ | \ | \ | & \to & 1\ 0\ 0 \\ & & \hline & & I\ I\ P \end{array}$$

c) En este apartado, veremos que una posición en Nim es ganadora si y sólo si es correcta.

i) Si ninguna fila tiene más de un fósforo, la posición es ganadora $\Leftrightarrow$ hay en total un número *par* de fósforos $\Leftrightarrow$ la posición es correcta.

ii) Si un número $a \in \mathbb{N}$, expresado en binario por los coeficientes $(a_n, \ldots, a_0)$ (permitiendo $a_n = 0$) es reemplazado por otro menor b (entero ≥ 0), expresado en binario por $(b_n, \ldots, b_0)$, entonces para algún i, $0 \leq i \leq n$, las paridades de a_i y b_i son distintas.

iii) Por lo tanto, si el jugador X recibe una posición correcta, necesariamente la transforma en incorrecta.

iv) Supongamos que estamos en una posición incorrecta, es decir, al menos la suma de una columna es impar. Para fijar ideas supongamos que las paridades de las columnas son

$$P \quad P \quad I \quad P \quad I \quad P$$

Entonces hay al menos un 1 en la tercera columna (la primera con suma impar). Supongamos, otra vez para fijar ideas, que una fila en la cual esto pasa es (en binario)

$$0 \quad 1 \quad \overline{1} \quad 1 \quad \overline{0} \quad 1$$

donde marcamos con $-$ que los números debajo están en columnas de suma impar. Cambiando 0's y 1's por 1's y 0's en las posiciones marcadas, obtenemos el número (menor que el original, expresado en binario):

$$0 \quad 1 \quad \overline{0} \quad 1 \quad \overline{1} \quad 1$$

Está claro que este cambio corresponde a un movimiento permitido, haciendo la suma de cada columna par, y que el argumento es general.

v) Por lo tanto, si el jugador X recibe una posición incorrecta, puede mover de modo de dejar una posición correcta.

vi) Si A deja una posición correcta, B necesariamente la convierte en incorrecta y A puede jugar dejando una posición correcta. Este proceso continuará hasta que cada fila quede vacía o contenga un único fósforo (el caso *i*)).

d) En base a los apartados anteriores, y suponiendo que A y B

siempre juegan de modo óptimo, ¿quién ganará?

e) Ver que la «jugada ganadora» en *c.iv)* no siempre es única.

f) Desarrollar una función que, ingresada una posición inicial en la forma de una lista con el número de fósforos en cada fila, decida si la posición es correcta o incorrecta, y en este caso encuentre una jugada ganadora.

g) ¿Cuál es la estrategia si el que saca el último fósforo es el que *pierde*?

✍ Tomado de Hardy y Wright (2008). ✂

5. Raíces de ecuaciones

Para muchos de los algoritmos que veremos, será conveniente tener un método para resolver ecuaciones no lineales y acá presentamos dos de los más usados en problemas de una variable: bisección y el método de Newton-Raphson.

Nos enfocamos en el problema de encontrar $\lfloor \sqrt{n} \rfloor$ para $n \in \mathbb{N}$, que será lo que más usaremos.

E 5.1 (método de la bisección). Consideremos un conjunto A de enteros consecutivos, $A = \{a, a+1, \ldots, b\}$, y supongamos que tenemos una función $f : A \to \mathbb{R}$ creciente, es decir, $f(k) < f(k+1)$ para $a \le k < b$.

Si $x \in \mathbb{R}$ es tal que $f(a) \le x < f(b)$, podemos encontrar n tal que $f(n) \le x < f(n+1)$ con el *método de la bisección*: se divide al conjunto A en dos partes y se elige la mitad en la cual podría estar x, luego se divide la mitad elegida en dos y se repite el procedimiento, y así sucesivamente.

Un esquema en Python sería:

```
# acá debemos saber que f(a) <= x < f(b)
poco = a
mucho = b
while poco + 1 < mucho:
    medio = (poco + mucho) // 2
    if x < f(medio):
        mucho = medio
    else:
        poco = medio
```

```
    # acá es f(poco) <= x < f(mucho)
# al final del lazo mucho <= poco + 1
return poco
```

a) Ver que el esquema es correcto, y en cada paso se mantiene el *invariante* $f(poco) \le x < f(mucho)$.

b) ¿Podría ser que `poco` fuera `mucho` al final del lazo propuesto?

c) ¿Por qué se pide `poco + 1 < mucho` y no `poco < mucho`?

d) Construir una función para calcular $\lfloor \sqrt{n} \rfloor$ cuando $n \in \mathbb{N}$ usando este método, donde $f(x) = x^2$, $a = 1$ y $b = n$ (si $n > 1$).

e) Comparar la función construida con `int(n**(1/2))` para valores bajos de n.

f) Cuando Python no puede representar el entero n como decimal exactamente, puede que el cálculo de `int(n**(1/2))` no sea exacto. Encontrar un valor de n donde esto suceda.

g) ¿Cuántos pasos se hacen en este método (con los valores iniciales propuestos) en términos de la cantidad de bits de n? ✂

E 5.2 (método «babilónico» para la raíz cuadrada). Este método para aproximar la raíz cuadrada del real positivo a, consiste en aplicar sucesivamente las iteraciones

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \quad \text{para } n = 1, 2, \ldots, \tag{5.1}$$

a partir de un valor inicial x_0 dado ($x_0 > 0$). Es decir $x_{n+1} = f(x_n)$, donde está definida sobre los reales no negativos como

$$f(x) = \frac{1}{2}\left(x + \frac{a}{x}\right).$$

a) $f(x) \ge \sqrt{a}$ para todo $x > 0$.

b) $f(x) > x$ si y sólo si $x < \sqrt{a}$, y $f(x) = x$ (x es un punto fijo de f) si y sólo si $x = \sqrt{a}$.

c) $f(x) - \sqrt{a} = \dfrac{(x - \sqrt{a})^2}{2x}$.

✍ Cuando x está cerca de $\sqrt{a}$, $f(x)$ está *mucho* más cerca, y decimos que la convergencia del método es *cuadrática*.

d) Si $x > \sqrt{a}$, $\left|f(x) - \sqrt{a}\right| < \frac{1}{2}\left|x - \sqrt{a}\right|$, i. e., acortamos más de la mitad la distancia a $\sqrt{a}$.

✍ Por lo tanto el método converge independientemente del punto inicial (siempre que sea positivo), y decimos que la convergencia es *global*.

e) Si $x_0 > 0$ y definimos x_n como en (5.1), entonces $x_1 \geq x_2 \geq \cdots \geq \sqrt{a}$, y si $x_n = \sqrt{a}$ para algún n, entonces $x_n = x_{n-1} = \cdots = x_0 = \sqrt{a}$.

Por lo tanto, si suponemos precisión "infinita" el método *nunca* da la respuesta exacta salvo que $x_0 = \sqrt{a}$.

f) ¿Qué pasa si $a = 0$?, ¿y si $x_0 = 0$?

g) ¿Qué pasa si $x_0 < 0$ y $a > 0$?

h) ¿Qué pasa si $a < 0$ y $x_0 \in \mathbb{R}$?

i) Construir una función para calcular $\sqrt{a}$ usando el método tomando valor inicial $x_0 = 1$ terminando en cuanto dos valores sucesivos difieran en menos de 10^{-7}.

Comparar los resultados con los obtenidos por `a**(1/2)` para varios valores de a.

✍ Los babilonios tomaron poder de la Mesopotamia (entre los ríos Tigris y Éufrates) alrededor de 2000 a. C., desalojando a los sumerios. Fueron los sumerios los que desarrollaron el sistema sexagesimal (en base 60) que nos llega a la actualidad en la división de horas en minutos y segundos, y los babilonios continuaron con el desarrollo llegando a un sistema que en algunos sentidos es más avanzado que el decimal que usamos.

Los babilonios también estudiaron la resolución de ecuaciones cuadráticas y cúbicas, llegando al equivalente del método que presentamos en esta sección, descripto en la tableta «Yale» fechada entre 1800 y 1650 a. C..

Mucho más tarde, Isaac Newton (1642–1727) desarrolló en 1669 un método para encontrar raíces de ecuaciones racionales, que tiene como caso particular al método babilónico. J. Raphson (1648–1715) publicó una mejora del método en 1690, dando una formulación similar a la que usamos ahora. No obstante, T. Simpson (1710–1761) fue el primero en aplicar el método a funciones trascendentes en 1740 (sin mencionar ni a Newton ni a Raphson ☹).

Las computadoras y calculadoras usan variantes del método de Newton-Raphson para calcular funciones como el seno o el logaritmo.

E 5.3. Si $n \in \mathbb{N}$, $z \in \mathbb{Z}$ y $x \in \mathbb{R}$, entonces

$$\left\lfloor \frac{z+x}{n} \right\rfloor = \left\lfloor \frac{z+\lfloor x \rfloor}{n} \right\rfloor .$$

E 5.4 (babilónico discreto). Aún cuando x no esté cerca de $\sqrt{a}$, si $x > \sqrt{a}$ la función f del método babilónico acorta al menos en dos la distancia a $\sqrt{a}$ (ejercicio 5.2.*d*)), como lo hace el método de la bisección (ejercicio 5.1).

En este ejercicio modificamos el método babilónico para calcular $\lfloor \sqrt{n} \rfloor$ cuando $n \in \mathbb{N}$ usando sólo operaciones enteras, cambiando la función f por

$$g(x) = \left\lfloor \frac{x + \lfloor n/x \rfloor}{2} \right\rfloor ,$$

de modo que para cualquier $x > 0$ vale $g(x) \le f(x)$.

✍ Recordar que si $a \in \mathbb{Z}$ y $b \in \mathbb{N}$, entonces $\lfloor a/b \rfloor$ es la división entera (1.2), por lo que el cálculo de $g(x)$ involucra sólo operaciones entre enteros (si $x \in \mathbb{Z}$, $x \neq 0$).

Para fijar ideas, tomamos como punto inicial $x_0 = \lfloor (n+1)/2 \rfloor$ y luego $x_{i+1} = g(x_i)$ para $k \ge 0$.

Demostrar los siguientes apartados teniendo en cuenta los ejercicios 5.2 y 5.3.

a) $g(x) = \lfloor f(x) \rfloor$.

b) Si $x_i > \sqrt{n}$ entonces $x_i > x_{i+1}$.

c) $x_i \ge \lfloor \sqrt{n} \rfloor$ para todo $i \ge 0$.

d) Existe i tal que $x_i = \lfloor \sqrt{n} \rfloor$.

e) A diferencia del caso continuo, es posible que exista k tal que $x_{k+1} > x_k$: encontrar los valores de n para lo cual esto es posible y ver que en este caso los valores de x_i oscilan para $i \ge k$: $x_k = x_{k+2} = x_{k+4} = \cdots$ y $1 + x_k = x_{k+1} = \cdots = x_{k+3} = \cdots$

Sugerencia: si $x_i = \lfloor \sqrt{n} \rfloor \le \sqrt{n} < x_i + 1$ y $n = x_i^2 + j$ con $0 \le j < 2x_i$, entonces $x_{i+1} = x_i$, pero si $n = x_i^2 + 2x_i$ entonces...

f) ¿Cómo saber si $x_k = \lfloor \sqrt{n} \rfloor$ (y no hacer más iteraciones) sin calcular x_i^2 en todas las iteraciones?

g) Definir una función para aplicar este método usando un esquema como:

```
x = (1 + n) // 2
y = (x + n // x) // 2
while y < x:
    x = y
    y = (x + n // x) // 2
return x
```

h) Encontrar un valor de n tal que `int(n**(1/2))` difiera del valor encontrado por la función del apartado anterior (como hicimos en el método de la bisección).

i) Comparar los métodos de la bisección y el actual. ¿Cuál parece mejor?

✍ El método de Newton-Raphson para encontrar las soluciones de $x^m = a$ en el caso continuo también puede modificarse para considerar el caso discreto. Sin embargo, para m grande no es claro que sea más ventajoso que el método de la bisección.

✍ La rapidez de ambos métodos (bisección o Newton-Raphson) depende en forma esencial de los valores iniciales. Es posible dar cotas sencillas para mejorar mucho el tiempo que tardan (especialmente para números con cientos o miles de cifras). ✂

E 5.5. En algunos algoritmos avanzados de factorización, dado el entero $n > 1$ se trata de encontrar enteros $a > 1$ y $k > 1$ tales que $a^k = n$.

Usando bisección, construir una lista de todos los pares (a,k) tal que $a^k = n$ (la lista será vacía si no existen a y k), haciendo del orden de $\log_2(n)$ pasos. ✂

6. Máximo común divisor

E 6.1 (menor factor). Sea a entero positivo.

a) Si b y c son enteros positivos tales que $a = b \times c$ entonces $b \le \sqrt{a}$ o $c \le \sqrt{a}$.

b) Construir una función que dado el entero $n > 1$ calcule el menor m tal que $m \mid n$ y $m > 1$, recorriendo los números $2, 3, \ldots$
La búsqueda debe realizarse en no más de $\lfloor \sqrt{n} \rfloor$ pasos.

✍ Se pretende ir recorriendo —más o menos eficientemente— los números mayores que 1 y probando si dividen a n.

Por ejemplo, recordando la sección 5 podríamos seguir un esquema parecido a:

```
s = raiz(n)          #  ⌊√n⌋
for d in range(2, s + 1):
    if n % d == 0:
        return d
return n
```

La cantidad de pasos se puede reducir básicamente a la mitad si primero probamos con 2 y luego seguimos con los impares:

```
if n % 2 == 0:
    return 2
for d in range(3, s + 1, 2):
    if n % d == 0:
...
```

c) Demostrar que el número m calculado en el apartado anterior es primo.

d) Construir una función `esprimo(a)` que decida si a es primo o no en base a las ideas anteriores. ✂

E 6.2 (divisores). Consideremos el conjunto D de divisores del número natural n,

$$D = \{d \in \mathbb{N} : d \mid n\}.$$

a) Definir una función para construir una lista ordenada de los elementos de D dado $n \in \mathbb{N}$.

✍ Bastará buscar divisores que no superen a $\lfloor \sqrt{n} \rfloor$.

b) ¿Cuáles son los enteros entre 1 y 144 con mayor cantidad de divisores?

✍ Esto nos habla de los posibles beneficios al trabajar con los números escritos en bases distintas de 10. ✂

E 6.3. Dados los enteros positivos a y b consideremos el conjunto $\mathscr{D}$ de divisores comunes,

$$\mathscr{D} = \{c \in \mathbb{N} : c \mid a \text{ y } c \mid b\}.$$

a) Demostrar que $\mathscr{D}$ tiene un máximo.

✍ Observar la similitud con el ejercicio 4.1.

Denotamos con $\mathrm{mcd}(a,b)$ (por *máximo común divisor*) a $\max \mathscr{D}$.

✍ Alternativamente se podría usar la notación mdc por *máximo divisor común*, que es más castizo pero en contra de las notaciones tradicionales.

b) Una manera tremendamente brutal de encontrar $\mathrm{mcd}(a,b)$ es tomar literalmente los divisores de a, los de b, y encontrar el máximo de la intersección.

Definir una función en Python con estas ideas, siguiendo un esquema como

```
divsa = divisores(a)
divsb = divisores(b)
# intersección:
coms = [x for x in divsa if x in divsb]
return max(coms)
```

donde `divisores` es la función construida en el ejercicio 6.2.

c) Sea $\mathscr{F}$ el conjunto de las combinaciones lineales enteras positivas de a y b,

$$\mathscr{F} = \{z \in \mathbb{N} : z = ua + vb \text{ con } u, v \in \mathbb{Z}\}.$$

Ver que $\mathscr{F}$ tiene un mínimo, que indicamos en lo que sigue por f.

d) Demostrar que $f \mid a$ y $f \mid b$ (es decir, $f \in \mathscr{D}$).

e) Demostrar que si $c \mid a$ y $c \mid b$ (es decir, si $c \in \mathscr{D}$), entonces $c \mid f$.

f) $\mathrm{mcd}(a,b) = \max \mathscr{D} = \min \mathscr{F}$.

La igualdad

$$\mathrm{mcd}(a,b) = ua + bv \tag{6.1}$$

con u y v enteros se llama *igualdad o identidad de Bézout*.

g) ¿Valen los apartados anteriores si en vez de *dos* enteros (a y b) consideramos *un conjunto* de números naturales? Concretamente, si A es un subconjunto no vacío de números naturales —finito o infinito— y definimos

$$\mathscr{D} = \{c \in \mathbb{N} : c \mid a \text{ para todo } a \in A\},$$

y

$$\mathscr{F} = \left\{ z \in \mathbb{N} : z = \sum_i u_i a_i \text{con } u_i \in \mathbb{Z} \text{ y } a_i \in A \right\},$$

donde las sumas involucradas son finitas, nos preguntamos si existen máx $\mathscr{D}$, mín $\mathscr{F}$ y en caso afirmativo si máx $\mathscr{D}$ = mín $\mathscr{F}$.

✍ E. Bézout (1730–1783) demostró la identidad para polinomios. Sin embargo, Bachet la había planteado para enteros con anterioridad.

✍ La función construida en el apartado *b)* es demasiado ineficiente, mientras que el estudio en *c)* es elegante y muy útil desde la teoría, pero no dice cómo construir el máximo común divisor.

En la sección 8 veremos un algoritmo eficiente para el cálculo. ✂

E 6.4. ¿Qué debería ser $\mathrm{mcd}(a,b)$ cuando a y b son enteros arbitrarios (considerando también la posibilidad de que uno o ambos se anulen)? ✂

E 6.5. Supongamos que a y b son enteros no nulos.

a) (Pero Grullo) $\mathrm{mcd}(a,b) = \mathrm{mcd}(b,a)$.

b) Si $a \mid (b \times c)$ y $\mathrm{mcd}(a,b) = 1$ entonces $a \mid c$.

c) Si $d = \mathrm{mcd}(a,b)$, $a' = a/d$ y $b' = b/d$ entonces

$$\mathrm{mcd}(a',b) = \mathrm{mcd}(a,b') = 1.$$ ✂

7. Primos y factorización

Desde el punto de vista teórico no hay muchos problemas en trabajar con números primos. Por ejemplo, en las escuelas muchas veces se usa[(3)] la descomposición en producto de primos para encontrar el máximo común divisor.

Sin embargo, desde el punto de vista práctico la cosa se complica: no se conocen algoritmos eficientes para encontrar factores no triviales de un número.

Esta dificultad es aprovechada por la criptografía para, por ejemplo, enviar datos bancarios por internet, y exploraremos un poco este tipo de aplicaciones en la sección 11.

¡Quien encuentre un algoritmo eficiente para factorizar enteros puede hacer colapsar al sistema bancario mundial!

(3) O usaba...

Curiosamente, decidir si un número es primo o no es mucho más sencillo: en 2002, Agrawal, Kayal y Saxena probaron que existe un algoritmo muy eficiente para este problema. No lo veremos aquí pues nos desviaríamos demasiado.

Nos contentaremos con hacer parte del estudio teórico relacionados con primos y factorización, y proponer algoritmos *extremadamente ineficientes* para decidir si un número es primo o factorizarlo, tal vez razonables para números con unas pocas decenas de dígitos pero seguramente demasiado inapropiados para trabajar con los cientos de dígitos que se manejan en criptografía.

Pero volvamos al trabajo, para el cual es conveniente repasar el ejercicio 6.5.

E 7.1 (teorema fundamental de la aritmética). Este teorema establece que todo entero n, $n > 1$, se puede expresar de manera única como producto de primos:

$$n = p_1 \times p_2 \times \cdots \times p_r, \tag{7.1}$$

o, contando la multiplicidad,

$$n = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \cdots \times p_k^{\alpha_k}. \tag{7.2}$$

donde $\alpha_1, \alpha_2, \ldots, \alpha_k$ son naturales.

a) ¿Cómo se demuestra este teorema?

b) Definir una función para encontrar la factorización en primos del entero $n > 1$, obteniendo los resultados de dos formas:

i) Una lista de primos $(p_1, p_2, \ldots, p_n)$, donde los primos pueden estar repetidos, de modo que vale (7.1).

✍ La función puede hacerse ligeramente más eficiente si se va actualizando n, por ejemplo con alguna variante del siguiente esquema:

```
ps = []              # lista de primos
while n % 2 == 0:    # caso 2
    n = n // 2
    ps.append(2)
m = 3                # los impares
while True:
    s = raiz(n)      #  ⌊√n⌋
```

```
    while (m <= s) and (n % m != 0):
        m = m + 2
        n = n // m
    # acá (m > s) o (m divide a n)
    if m > s:       # n es primo
        ps.append(a)
        return ps   # salir
    # acá m es primo y divide a n
    while True:         # repetir
        ps.append(m)
        n = n // m
        if n % m != 0:  # hasta que
            break       # m no divide a n
    if n == 1:      # no queda nada
        return ps   # salir
    m = m + 2       # nuevos n y m
```

ii) Una lista de la forma $((p_1,\alpha_1),(p_2,\alpha_2),\ldots,(p_k,\alpha_k))$, de modo que vale (7.2).

c) Usar el apartado anterior para encontrar $\mathrm{mcd}(a,b)$ usando la descomposición en primos de a y b.

✍ Gauss fue quien primero enunció en forma clara este teorema en sus *Disquisitiones Arithmeticae* de 1801.

✍ El esquema propuesto en *b.i*) tarda del orden de $\sqrt{n}$ pasos cuando n es primo (o tiene sólo dos factores de tamaño similar). Es decir, la cantidad de pasos es exponencial en la cantidad de bits (el tamaño) de n. ✂

E 7.2 (criba de Eratóstenes). Recordando que los índices en Python empiezan en 0, el siguiente es un esquema para encontrar los números primos que no superan al entero n basado en la *criba de Eratóstenes*:

```
if n <= 1:
    return []

if n == 2:
    return [2]

esprimo = [True for k in range(n+1)]
```

```
    # incluimos 0 y 1 por comodidad

# los múltiplos de 2
for x in range(4, n+1, 2):
    esprimo[x] = False

# sólo impares
s = raiz(n)     # piso de la raíz cuadrada
for p in range(3, s + 1, 2):
    if esprimo[p]:
        for k in range(p * p, n + 1, 2 * p):
            esprimo[k] = False

return [p for p in range(2, n+1) if esprimo[p]]
```

a) Ver que el esquema es correcto.

b) Construir una función con estas ideas y verificar su comportamiento para $n = 10$ y $n = 100$.

E 7.3 (teorema de los números primos). Indiquemos por $\pi(n)$ a la cantidad de primos que no superan n.

Según cuenta el mismo Gauss, tenía 15 o 16 años (hacia 1797 o 1798) cuando conjeturó que

$$\pi(n) \approx \frac{n}{\log n} \quad \text{para } n \text{ grande,} \tag{7.3}$$

donde log es el logaritmo natural, aproximación que fue demostrada en 1896 en forma independiente por Jacques Hadamard y Charles Jean de la Vallée Poussin.

Verificar la bondad de la aproximación (7.3) para $n = 100$, $1\,000$, $10\,000$ y $100\,000$.

E 7.4 (distribución de los primos). Encontrar cuántos números primos que no superan n terminan en 1, 3, 7 y 9 para n grande ($10\,000$ o más). ¿Qué conjetura harías?

✍ Si bien se sabe que hay infinitos primos desde Euclides, recién hacia fines del siglo XIX pudieron resolverse algunas cuestiones muy generales sobre cómo se distribuyen.

Johann Peter Gustav Lejeune Dirichlet (1805–1859) demostró en 1837 que toda progresión aritmética $a+bk$, $k=1,2,\ldots$, contiene infinitos primos si a y b son coprimos. Dicho de otra forma, si $\mathrm{mcd}(a,b)=1$, hay infinitos primos que son congruentes con a módulo b.

Si pensamos que $b=10$, y a es cualquier número que no tenga como factores a 2 ni a 5, el teorema de Dirichlet dice que hay infinitos primos que terminan en 1, infinitos que terminan en 3, etc., pero no dice cómo es la distribución. Esto fue demostrado en forma general por Franz Carl Joseph Mertens (1840–1927) al refinar los resultados de Dirichlet. En nuestro caso, los resultados de Mertens aseguran que la cantidad de primos terminados en 1, 3, 7 o 9 es asintóticamente igual.

Un poco como contrapartida al teorema de Dirichlet, Ben Green y Terence Tao demostraron en 2004 que los números primos contienen progresiones aritméticas arbitrariamente largas. ✂

E 7.5 (complejidad de la criba de Eratóstenes). Los algoritmos propuestos en los ejercicios 6.1 o 7.1 para encontrar —respectivamente— el menor factor o descomponer en primos el entero n tardan del orden de $\sqrt{n}$ pasos (en el peor caso), es decir, son exponenciales en el tamaño de la entrada.

Esos algoritmos no difieren demasiado de la propuesta del ejercicio 7.2 para la criba de Eratóstenes, pero las salidas (y sus tamaños) son completamente diferentes.

En este ejercicio estudiamos la complejidad de la criba de Eratóstenes (según la describimos en el ejercicio 7.2), para lo cual conviene recordar el estudio de la función `factorial` en el ejercicio 3.6.

a) Estimar la cantidad de pasos que se realizan en términos del tamaño de la entrada (n) y la salida y comparar con el tamaño de la salida (ver ejercicio 7.3).

✍ A medida que n crece (el «límite para n tendiendo a infinito»),

$$\sum_{k=1}^{n} \frac{1}{k} \approx \gamma + \log(n),$$

donde $\gamma = 0.57721566\ldots$ es la *constante de Euler-Mascheroni* y log es el logaritmo natural (base e).

Si nos limitamos a sumar los inversos multiplicativos de

primos, se obtiene

$$\sum_{\substack{p \text{ primo} \\ p \le n}} \frac{1}{p} \approx M + \log(\log(n)),$$

donde $M = 0.26149721\ldots$ es la *constante de Meissel-Mertens*.

En fin,

$$\sum_{k=1}^{n} \log k = \log(n!),$$

que, por la fórmula de Stirling, es del orden de $\frac{1}{2}\log(2\pi) - n + (n + \frac{1}{2})\log n$ para n grande.

b) Estimar la cantidad de memoria (en bits) usada antes de construir la lista final de primos. ✂

E 7.6 (primos gemelos). Dos primos p y q se llaman *gemelos* si $|p - q| = 2$.

Encontrar todos los pares de primos gemelos (p, q) con $p < q \le 10\,000$.

✍ No se sabe si hay infinitos pares de primos gemelos. ✂

E 7.7 (conjetura de Goldbach). Esta conjetura, originada en la correspondencia entre el matemático ruso Christian Goldbach (1690–1764) y Euler en 1742, dice que todo número par mayor que 4 puede escribirse como suma de dos números primos impares (no necesariamente distintos).

En este ejercicio haremos pruebas computacionales relacionadas, donde suponemos n par, $n \ge 6$.

a) Decidir si el siguiente esquema se puede tomar como base para encontrar primos impares p y q tales que $n = p + q$, y en caso contrario decir cuáles son los inconvenientes y cómo podrían resolverse.

```
c = criba(n)    # criba de Eratóstenes
i = 1           # c[0] = 2, c[1] = 3
j = len(c) - 1
while i <= j:
    s = c[i] + c[j]
    if s == n:
        return (c[i], c[j])     # p, q
```

```
    elif s > n:
        j = j - 1
    else:
        i = i + 1
# n no puede escribirse como p + q
```

b) Construir una función para decidir si n se puede escribir como suma de dos primos impares, y luego verificar la conjetura para todos los números pares que no superan 10 000.

c) Construir una función para, dado n par y mayor que 4, hacer una lista de todas los primos p impares, tales que $n-p$ es primo impar, y $2p \le n$.

Por ejemplo

```
>>> goldbach(84)
[5, 11, 13, 17, 23, 31, 37, 41]
```

d) La *cuenta de Goldbach* es la cantidad de formas en las que un número par n puede escribirse como $n = p + q$, donde p y q son primos impares, $p \le q$. Hacer un programa para determinar la cuenta de Goldbach de todos los números pares entre 100 y 10 000, encontrando el mínimo y el máximo.

✍ Ver la sucesión A002375 en https://oeis.org.

✍ No se sabe si la conjetura es cierta.

✍ Goldbach consideraba que 1 era primo, en cuyo caso la conjetura cambia un poco.

✍ Hay muchas variantes de la conjetura. Posiblemente la más conocida sea la que todo impar mayor que 7 es suma de tres primos impares, aparentemente demostrada por H. Helfgott en 2013 (pero hasta fines de 2018 no había sido publicada en una revista reconocida).

✍ Comparar el enunciado de la conjetura con el del ejercicio 7.13. ✂

E 7.8 (primos de Euclides). La demostración de la infinitud de números primos de Euclides sugiere considerar números de la forma

$$1 + p_1 \times p_2 \times \ldots p_n,$$

donde $p_1, p_2, \ldots, p_n$ son los n primeros primos, que llamamos *números de Euclides*.

Ver que hay números de Euclides que no son primos, y encontrar todos los números de Euclides menores que 10 000 que sí lo son.

✍ No se sabe si hay infinitos primos de esta forma, ni siquiera se sabe si todo número de Euclides es libre de cuadrados.

✍ La demostración de Euclides no presupone que se estén tomando los primeros primos, sólo un conjunto finito (y se ve que hay otro primo fuera de ese conjunto). ✂

E 7.9 (primos de Fermat). Sean n y m enteros positivos.

a) Si $2^n + 1$ es primo entonces n debe ser potencia de 2.

b) Sea $F_n = 2^{2^n} + 1$ (el n-ésimo número de Fermat).
Probar que si $n \neq m$ entonces $\mathrm{mcd}(F_n, F_m) = 1$.

c) Ver que el apartado anterior es la base de otra demostración de la existencia de infinitos primos.

d) Fermat conjeturó que todos los F_n son primos. Ver que esa conjetura es falsa calculando los primeros números de Fermat.

✍ No se sabe si hay infinitos primos de Fermat, y sólo se conocen cinco de ellos: 3 ($n = 0$), 5, 17, 257 y 65537 ($n = 4$).

✍ Gauss demostró que se puede construir con regla y compás un polígono regular de n lados si $\phi(n)$ (ver ejercicio 9.19) es una potencia de 2, esto es, si n es el producto de una potencia de 2 y un primo de Fermat. P. Wantzel demostró en 1837 que la condición también es necesaria.

Comparar esta condición con la relación entre los primos de Mersenne y los números perfectos (ejercicios 7.10 y 9.23). ✂

E 7.10 (primos de Mersenne). Dado $n \in \mathbb{N}$, llamamos *número de Mersenne de orden n* al número $M_n = 2^n - 1$.

a) ¿Cuántos dígitos binarios tiene M_n?

b) Si M_n es primo entonces n es primo.

c) Hay relativamente pocos números de Mersenne que son primos:

i) Encontrar todos los M_n menores que 100 000 que son primos.

ii) Encontrar el menor M_n que no lo es.

✍ No se sabe si hay infinitos primos de Mersenne.

✍ Los «records» de números primos grandes en general se hacen usando primos de Mersenne, inclusive hay una red de computadoras tratando

de superar al anterior. El más grande encontrado hasta diciembre de 2018 es $M_{82589933}$ que tiene 24 862 048 dígitos decimales.
Ver https://www.mersenne.org.

✍ Ver la relación con números perfectos en el ejercicio 9.23. ✂

E 7.11 (período de una fracción). De la escuela recordamos que todo número racional p/q $(0 < p < q)$ tiene una «expresión decimal periódica». Por ejemplo,

- $1/7 = 0.1428571428\ldots = 0.\overline{142857}$ tiene período 142857,
- $617/4950 = 0.124646\ldots = 0.12\overline{46}$ tiene período 46 y anteperíodo 12,
- $1/4 = 0.25 = 0.2500\ldots = 0.25\overline{0}$ tiene período 0 y anteperíodo 25.

En este ejercicio vamos a construir una función que escribe la parte decimal (del desarrollo en base 10) del racional a/b (a y b enteros positivos), distinguiendo su período y anteperíodo, y luego otra función para recuperar a y b a partir del período y anteperíodo.

a) ¿Por qué los desarrollos decimales de los racionales son eventualmente periódicos?

b) Definir una función `aperiodo(a, b)` que dados $a, b \in \mathbb{N}$, con $0 < a < b$, retorne el anteperíodo y el período de la expresión decimal de a/b.
Por ejemplo `aperiodo(617, 4950)` debe dar por resultado `[[1, 2], [4. 6]]`, y `aperiodo(1, 7)` debe dar `[[], [1, 4, 2, 8, 5, 7]]`.
Comprobar el comportamiento con las fracciones mencionadas al principio del ejercicio.

c) Definir una función `deperiodo(ante, per)` que dadas la lista de cifras del anteperíodo en `ante` y la lista de cifras del período en `per` retorne los enteros a y b tales que a/b tiene ese anteperíodo y período.

d) Encontrar el entero n entre 1 y 1000 tal que $1/n$ tiene máxima longitud del período.
Respuesta: 983, que tiene período de longitud 982.

e) Encontrar todos los enteros entre 2 y 1000 para los cuales $1/n$ tiene período $n-1$. ¿Son todos primos?, ¿qué conjeturarías?, ¿podrías

demostrar tus conjeturas? ✂

E 7.12 (ternas pitagóricas). Una terna (x,y,z) de números naturales se dice *pitagórica* si

$$x^2+y^2=z^2, \tag{7.4}$$

siendo $(3,4,5)$ la más conocida. Desde ya que estas ternas están relacionadas con los triángulos rectángulos de lados enteros.

En este ejercicio estudiaremos algunas propiedades de estas ternas y cómo construirlas, viendo que hay infinitas soluciones.

Mucho de lo que haremos depende de la observación que la ecuación (7.4) es equivalente a $x^2=(z-y)(z+y)$.

a) Si (x,y,z) es una terna pitagórica, no puede ser $x=y$, y debe ser $\max\{x,y\}<z$ y $\min\{x,y,z\}\geq 3$.

b) Si (x,y,z) es una terna pitagórica, ¿podría ser $y=z-1$ y x par?

Recíprocamente, si x es impar, ¿pueden encontrarse siempre y y z con $z=y+1$ de modo que (x,y,z) sea una terna pitagórica?

c) Hacer una función `ternas1(n)` para encontrar todas las ternas (x,y,z) donde $y+1=z\leq n$.

d) Si (x,y,z) es una terna pitagórica y $m\in\mathbb{N}$, entonces (mx,my,mz) es una terna pitagórica.

e) Si (x,y,z) es una terna pitagórica y $d\in\mathbb{N}$ divide a dos de los tres números, entonces debe dividir al tercero.

En particular, $(x,y,z)=(mx',my',mz')$ para algún $m\in\mathbb{N}$, donde (x',y',z') forman una *terna primitiva*, es decir, donde los números son coprimos dos a dos.

f) Si (x,y,z) es una terna pitagórica, x o y debe ser par (ambos podrían serlo).

¿Debe ser alguno de x o y múltiplo de 3?

g) Si u y v son números naturales con $u>v$, entonces

$$x=2uv,\quad y=u^2-v^2,\quad z=u^2+v^2, \tag{7.5}$$

forman una terna pitagórica (en la que al menos x es par).

h) Si (x,y,z) es una terna primitiva y x es par, entonces existen u y v de distinta paridad tales que $u>v$, $\mathrm{mcd}(u,v)=1$ y vale (7.5).

Recíprocamente, si u y v son coprimos de distinta paridad y $u>v$, entonces la terna definida por (7.5) es primitiva.

i) Si r y s son números naturales con $r > s$ y ambos impares, entonces

$$x = \frac{r^2 - s^2}{2}, \quad y = rs, \quad z = \frac{r^2 + s^2}{2}, \tag{7.6}$$

son enteros y forman una terna pitagórica (en la que x es par y y y z son impares).

j) Si (x, y, z) es una terna primitiva y x es par, entonces existen r y s impares tales que $r > s$, $\mathrm{mcd}(r, s) = 1$ y vale (7.6).

Recíprocamente, si r y s son impares, $r > s$ y $\mathrm{mcd}(r, s) = 1$, entonces la terna definida por (7.6) es primitiva.

¿Qué relación hay entre u y v de la ecuación (7.5) y s y t en (7.6)?

k) Hacer una función `ternasp(n)` para encontrar todas las ternas primitivas donde x sea par y $z \le n$.

l) Hacer una función `ternas(n)` para encontrar todas las ternas pitagóricas donde $x < y$ y $z \le n$.

✍ Las ternas pitagóricas son conocidas desde la antigüedad, hay registros en una tableta babilónica de alrededor de 1800 a. C..

✍ Pitágoras de Samos (569 a. C.–500 a. C.) obtuvo la familia infinita de soluciones

$$x = 2t, \; y = t^2 - 1, \; z = t^2 + 1 \quad \text{con } t \in \mathbb{N}.$$

✍ La caracterización (7.5) está en el Libro X, Proposición 29 de *Los Elementos*.

E 7.13. En este ejercicio estudiamos una variante de las ternas pitagóricas, tratando de encontrar, dado $n \in \mathbb{N}$, enteros a y b tales que $n = a^2 + b^2$.

a) Demostrar la *identidad de Diofanto*:

$$\left(a^2 + b^2\right)\left(c^2 + d^2\right) = (ac - bd)^2 + (ad + bc)^2 .$$

b) A. Girard en 1625 y P. Fermat en 1640 enunciaron (sin dar demostración) que si p es primo, entonces $p \equiv 1 \pmod 4$ si y sólo sí p puede escribirse como suma de dos cuadrados, $p = a^2 + b^2$, resultado que fue demostrado recién en 1747 por Euler.

Definir una función `sumacuadp(n)` que dado $n \in \mathbb{N}$ decida si n es primo y $n \not\equiv 1 \pmod 4$, y en ese caso encuentre a y b tales que $n = a^2 + b^2$.

Aclaración: si bien se puede hacer en forma más elaborada, acá sólo se pide hacer un «barrido a la bruta» recorriendo valores posibles de a y b, como hicimos en los ejercicios 6.1 o 7.7.

c) Usando la caracterización de Girard-Fermat, demostrar que si $n \in \mathbb{N}$, entonces existen enteros a y b tales que $n = a^2 + b^2$ si y sólo si los factores primos impares de n que tienen resto 3 al dividirlos por 4 aparecen a una potencia par.

Por ejemplo, $1350 = 2\,3^3\,5^2$ no puede escribirse como suma de cuadrados y $1377 = 3^4\,17$ sí (e. g., $9^2 + 36^2$).

d) Definir una función `sumacuad(n)` que dado $n \in \mathbb{N}$ decida si n puede escribirse como suma de dos cuadrados (eventualmente alguno nulo) y en caso afirmativo encontrar enteros a y b tales que $n = a^2 + b^2$.

✍ En 1770 Lagrange demostró que todo entero positivo es suma de cuatro cuadrados (algunos eventualmente nulos). ✂

8. El algoritmo de Euclides

E 8.1 (Euclides según Euclides). En la Proposición 1 del Libro VII de *Los Elementos*, Euclides enuncia:

> *Dados dos números (enteros y positivos) distintos, y restando sucesivamente y continuamente el menor del mayor, si el número que queda nunca mide el anterior a él hasta que queda una unidad, los números originales serán primos entre sí.*

y en la Proposición 2:

> *Dados dos números y no primos entre sí, encontrar su máxima medida común.*

En lenguaje moderno, la primera nos dice que si restando sucesivamente el menor del mayor (modificando el mayor) llegamos a la unidad, entonces los números originales son coprimos. La segunda proposición explica que podemos encontrar el máximo común divisor, «la máxima medida común», haciendo sucesivas restas hasta llegar a la igualdad.

Podríamos esquematizar en Python el algoritmo propuesto por Euclides como:

```
# restar sucesivamente el menor del mayor
# hasta que sean iguales
while a != b:
    if a < b:
        b = b - a
    else:
        a = a - b
# acá a = b = mcd
```

a) Ver que el algoritmo es correcto, es decir:

- El algoritmo termina en un número finito de pasos.
- Respecto de los valores originales de a y b, al final del algoritmo efectivamente el valor obtenido es el de $\mathrm{mcd}(a,b)$ (es *divisor* de ambos y es *máximo*).

b) Construir una función implementando estas ideas, incluyendo los casos donde a o b pueden anularse o ser negativos, y verificar su corrección con diversas entradas.

✍ En el Libro X, Proposición 3 de *Los Elementos*, Euclides enuncia un resultado similar cuando los números (positivos pero no necesariamente enteros) son *conmensurables* entre sí.

Recordemos que un poco antes de Euclides con Pitágoras y el descubrimiento de la irracionalidad de $\sqrt{2}$, surgió el problema de la *conmensurabilidad* de segmentos, es decir, si dados dos segmentos de longitudes a y b existe otro de longitud c tal que a y b son múltiplos enteros de c (en cuyo caso c es una «medida común»). Si a es irracional (como $\sqrt{2}$) y $b = 1$, entonces no existe c, y el algoritmo de Euclides no termina nunca. ✂

E 8.2 (algoritmo de Euclides con división). Al restar sucesivamente el menor del mayor esencialmente estamos haciendo una división (ejercicio 4.2), excepto que en la versión original de Euclides nos detenemos si llegamos a la igualdad, mientras que con la división haríamos una resta más llegando a resto 0.

Así, si a y b son enteros positivos, al traducir el algoritmo original de Euclides de restas sucesivas a divisiones sucesivas, obtendríamos la serie de igualdades

$$\begin{aligned} a &= q_1 b + r_1 && \text{(restar } q_1 \text{ veces } b \text{ de } a\text{)}, \\ b &= q_2 r_1 + r_2 && \text{(restar } q_2 \text{ veces } r_1 \text{ de } b\text{)}, \\ &\vdots \\ r_{k-2} &= q_k r_{k-1} + r_k && \text{(restar } q_k \text{ veces } r_{k-1} \text{ de } r_{k-2}\text{)}, \\ r_{k-1} &= q_{k+1} r_k, \end{aligned} \tag{8.1}$$

donde en el último paso restamos q_{k+1} veces r_k de r_{k-1} llegando exactamente a r_k, o sea, $r_{k+1} = 0$, y terminamos.

a) Ver que el algoritmo es correcto, es decir, termina en un número finito de pasos, y que el último resto no nulo es $\mathrm{mcd}(a, b)$.

b) ¿Qué pasa cuando a o b son nulos o negativos?

c) Modificando adecuadamente el esquema

```
while b > 0:
    a, b = b, a % b
# acá a = mcd y b = 0
```

hacer una función para calcular $\mathrm{mcd}(a, b)$ usando divisiones, teniendo en cuenta los casos donde a o b sean nulos o negativos. ✂

E 8.3. *a*) Comparar las funciones obtenidas en los ejercicios 8.1 y 8.2, viendo que dan los mismos resultados (¡correctos!).

Usando ambas funciones, calcular $\mathrm{mcd}(612, 456)$.

b) Como ya mencionamos, Euclides también enuncia el algoritmo para números no enteros.

¿Cuál debería ser el resultado de $\mathrm{mcd}(6.12, 4.56)$?

c) ¿Cuál es el resultado de las funciones de los ejercicios 8.1 y 8.2 aplicadas a 6.12 y 4.56?, ¿por qué? ✂

E 8.4 (Euclides binario). Las computadoras pueden dividir o multiplicar por 2 o determinar si un número es par o impar muy rápidamente, usando operaciones (en general de bajo nivel) del tipo "shift", que no veremos aunque están implementadas en Python.

En este ejercicio vemos una variante del algoritmo de Euclides con esas ideas, donde se mezclan divisiones por 2 con restas.

a) Demostrar las siguientes propiedades cuando $m, n \in \mathbb{N}$.

i) $\mathrm{mcd}(2m,2n) = 2\,\mathrm{mcd}(mn)$.

ii) Si n es impar, $\mathrm{mcd}(2m,n) = \mathrm{mcd}(m,n)$.

iii) Si $m > n$, $\mathrm{mcd}(m,n) = \mathrm{mcd}(m-n,n)$.

iv) $\mathrm{mcd}(m,n) = \mathrm{mcd}(n,m)$.

v) $m = n \Rightarrow \mathrm{mcd}(m,n) = n$.

vi) Si m y n son impares, $m-n$ es par.

b) Hacer una función para calcular $\mathrm{mcd}(a,b)$ con estas ideas.

E 8.5. Calcular el máximo común divisor entre

$$351\,261\,949\,273\,842\,158\,506\,397 \text{ (24 dígitos)}$$

y

$$24\,195\,425\,306\,258\,454\,592\,079\,257 \text{ (26 dígitos)},$$

usando la descomposición en primos y el algoritmo de Euclides, viendo que se obtienen los mismos resultados.

✍ La idea es «copiar» de la versión electrónica del texto y luego «pegar» en la terminal de Python.

✍ No estaría de más usar las distintas versiones del algoritmo de Euclides: ejercicios 8.1, 8.2 y 8.4.

Respuesta: 306 244 070 857 752 535 751 que tiene 21 dígitos.

E 8.6 (complejidad del algoritmo de Euclides). Acá vemos que la versión del algoritmo de Euclides que vimos en el ejercicio 8.2 es muy eficiente.

Una primera observación, y que habla de que se termina en un número finito de pasos, es observar que los restos al menos decrecen en 1 en cada paso, por lo que la cantidad de pasos está dominada por $\min\{a,b\}$.

En realidad es mucho más eficiente.

Si f_n es el n-ésimo número de Fibonacci (ejercicio 3.7), revirtiendo las igualdades en (8.1) podemos poner

$$\begin{aligned} r_k &\geq 1 = f_2, \\ r_{k-1} = q_{k+1} r_k &\geq 2 \times 1 = f_3 \quad (\text{pues } r_{k-1} \neq r_k), \\ r_{k-2} = q_k r_{k-1} + r_k &\geq r_{k-1} + r_k \geq f_3 + f_2 = f_4, \\ &\vdots \end{aligned}$$

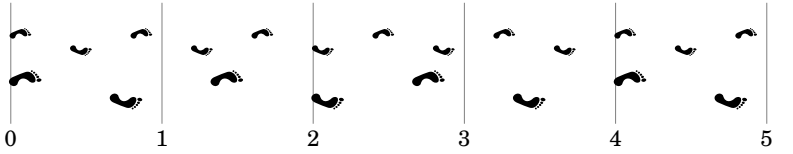


Figura 8.1: Pasos de Pablito y su papá.

y observando que los subíndices de r bajan en 1 mientras los de f suben en 1 en cada paso (sumando siempre $k+2$), llegamos a

$$r_1 = q_3 r_2 + r_3 \geq f_k + f_{k-1} = f_{k+1},$$
$$b = q_2 r_1 + r_2 \geq f_{k+1} + f_k = f_{k+2},$$
$$a = q_1 b + r_1 \geq f_{k+2} + f_{k+1} = f_{k+3}.$$

a) Si $a, b \in \mathbb{N}$, $a > b > 0$, y el algoritmo de Euclides (versión restos) realiza n divisiones, entonces $a \geq f_{n+2}$ y $b \geq f_{n+1}$.

b) Ver que el resultado anterior no se puede mejorar tomando $a = f_{n+2}$ y $b = f_{n+1}$.

c) Usando la fórmula de Euler-Binet (3.2) y logaritmos, demostrar el teorema de Lamé: el número de divisiones nunca supera 5 veces el número de cifras decimales del número menor.

En particular, la cantidad de operaciones es del orden del tamaño de las entradas. ✂

E 8.7 (Pablito y su papá I). Pablito y su papá caminan juntos tomados de la mano. Pablito camina 2 metros en exactamente 5 pasos, mientras que su padre lo hace en exactamente 3 pasos.

a) Resolver con lápiz y papel: si empiezan a caminar juntos, ¿cuántos metros recorrerán hasta marcar nuevamente el paso juntos?, ¿y si el padre caminara $2\frac{1}{2}$ metros en 3 pasos?

Aclaración: se pregunta si habiendo en algún momento apoyado simultáneamente los pies izquierdos, cuántos metros después volverán a apoyarlos simultáneamente (ver figura 8.1).

Respuesta: 4 y 20 metros respectivamente.

b) ¿Qué relación hay entre el máximo común divisor o el mínimo común múltiplo y el problema de Pablito y su papá?

✍ Otros ejemplos «clásicos» para introducir el mínimo común múltiplo son:

- En un engranaje con dos ruedas de distinto tamaño se pinta el segmento que une los centros cuando las ruedas están quietas, y se quiere ver cuántas vueltas da cada una hasta que vuelva a verse el segmento como uno solo.
- Los viajantes de comercio que recorren pueblos en tiempos distintos, y hay que ver cuándo vuelven a encontrarse en determinado pueblo.
- Los semáforos que cambian de luces con determinadas frecuencias y hay que ver cuándo vuelven a ponerse en el mismo color. ✂

E 8.8 (mínimo común múltiplo). El *mínimo común múltiplo* de $a, b \in \mathbb{N}$, que indicamos con $\text{mcm}(a,b)$, se define en forma análoga al máximo común divisor: es el menor entero del conjunto $\{k \in \mathbb{N} : a \mid k \text{ y } b \mid k\}$.

a) En la escuela nos enseñan[(4)] que si $a, b \in \mathbb{N}$ entonces $\text{mcm}(a,b) \times \text{mcd}(a,b) = a \times b$. ¿Por qué es válida esta relación?

b) ¿Cómo podría extenderse la definición de $\text{mcm}(a,b)$ para $a, b \in \mathbb{Z}$?, ¿cuál sería el valor de $\text{mcm}(0,z)$? ✂

E 8.9 (Pablito y su papá II). Definir una función para resolver en general el problema de Pablito y su papá (ejercicio 8.7), donde las entradas son el número de pasos y la cantidad de metros recorridos tanto para Pablito como para su papá.

Concretamente, los argumentos de la función son los enteros positivos p_b, n_b, d_b, p_p, n_p y d_p, donde:

p_b : número de pasos de Pablito
n_b/d_b : metros recorridos por Pablito en p_b pasos
p_p : número de pasos del papá
n_p/d_p : metros recorridos por el papá en p_p pasos

✍ Recordando el tema de la conmensurabilidad mencionado al introducir el algoritmo de Euclides, no siempre el problema tiene solución. Por ejemplo, si Pablito hace 1 metro cada 2 pasos y el papá $\sqrt{2}$ metros cada 2 pasos.

Como para la computadora todos los números son racionales, el problema siempre tiene solución computacional. ✂

(4) ¡A veces!

E 8.10 (visibilidad en el plano). Consideremos el *reticulado* $\mathscr{C}$ definido por los puntos del plano con ambas coordenadas enteras. Decimos que el punto $P \in \mathscr{C}$ es *visible* (desde el origen O) si el segmento que une P y O no contiene otros puntos de $\mathscr{C}$ distintos de P y O.

a) Si $a, b \in \mathbb{N}$, entonces (a, b) es visible si y sólo si $\text{mcd}(a, b) = 1$.

b) Desarrollar una función que dado $n \in \mathbb{N}$ calcule s_n, la cantidad de puntos visibles en el cuadrado $1 \le a, b \le n$.

c) Se puede demostrar que a medida que n crece, $p_n = s_n/n^2$ se aproxima a $p = 6/\pi^2 = 0.6079\ldots$. Calcular p_n para distintos valores de n para ver que este es el caso.

Sugerencia: Empezar con n pequeño e ir aumentando paulatinamente su valor.

✍ Tomado de Engel (1993). ✂

E 8.11 (el juego de Euclides). En un tablero, cuyas casillas suponemos identificadas como (x, y), con x, y enteros no negativos, se coloca una ficha en la posición (a, b) con $a, b \neq 0$. Dos jugadores juegan alternativamente, moviendo la ficha desde la posición (x, y), a una casilla de la forma $(x - ty, y)$ o $(x, y - tx)$, según si $x \ge y$ o $y \ge x$, donde $t \in \mathbb{N}$ es elegido por el jugador pero es tal que las nuevas coordenadas son ambas no negativas. Gana el jugador que primero llegue a una posición en la que alguna de las coordenadas es 0.

a) Ver que cualquier sucesión de movimientos que empieza en (a, b) debe terminar en el par $(0, \text{mcd}(a, b))$ (o su simétrico).

b) Si $y < x \le y \times (1 + \sqrt{5})/2$, hay un único movimiento posible desde (x, y). Además, si se llega a (z, y), debe ser $y > z \times (1 + \sqrt{5})/2$.

c) Ver que si un juego empieza en (a, b) y $a \ge b$, el primer jugador tiene una estrategia ganadora si $a = b$ o si $a > b \times (1 + \sqrt{5})/2$. En caso contrario, el segundo jugador tiene una estrategia ganadora.

d) Desarrollar un programa para jugar el juego de Euclides en el que:

i) La posición inicial es entrada por el jugador humano.

ii) Luego juegan alternativamente la computadora (óptimamente) y el usuario.

iii) La computadora muestra su movimiento por pantalla y el usuario ingresa su movida por teclado.

✍ En Python esto puede hacerse con la función `input` que lee la entrada como cadena de caracteres (`str`) y hay que convertirla a entero o decimal:

```
a = int(input("Entrar un número: "))
print("El número ingresado es:", a)
```

iv) La computadora verifica que el movimiento propuesto por el usuario es válido, y vuelve a requerir una entrada válida en caso contrario.

✍ Tomado de Rosen (1993). ✂

9. El algoritmo de Euclides extendido

Mirando las ecuaciones (8.1), vemos que poniendo por comodidad

$$r_0 = b, \quad r_{-1} = a, \tag{9.1}$$

podemos expresar a r_j como combinación lineal de los dos restos anteriores,

$$r_i = r_{i-2} - q_i r_{i-1} \quad \text{para } j \geq 1. \tag{9.2}$$

Yendo «de atrás para adelante», r_k es combinación lineal de r_{k-1} y r_{k-2}, como r_{k-1} es combinación lineal de r_{k-1} y r_{k-3}, r_k puede escribirse como combinación lineal de estos dos últimos, y así sucesivamente, vemos que se puede escribir $r_k = \text{mcd}(a,b)$ como combinación lineal de a y b.

Concretamente, si $d = \text{mcd}(a,b)$ y sabemos que

$$d = s_i r_i + t_i r_{i-1}, \tag{9.3}$$

para algún i con $1 \leq i \leq k$, por (9.2) tenemos

$$\begin{aligned} d &= s_i r_i + t_i r_{i-1} = s_i \left(r_{i-2} - q_i r_{i-1}\right) + t_i r_{i-1} \\ &= \left(t_i - s_i q_i\right) r_{i-1} + s_i r_{i-2}, \end{aligned}$$

por lo que

$$s_{i-1} = t_i - s_i q_i, \quad t_{i-1} = s_i, \tag{9.4}$$

obteniendo (9.3) para i reemplazado por $i-1$. Claro que cuando lleguemos a $i = 0$, obtendremos la identidad de Bézout (6.1).

E 9.1. Hacer una función en Python para encontrar u y v para que valga (6.1), siguiendo los pasos descriptos anteriormente, guardando los coeficientes q_i para $i = 1, \ldots, k$ en la «pasada hacia adelante» en (8.1), y usando (9.4) y (9.1) para la «pasada hacia atrás». ✂

El algoritmo propuesto en el ejercicio 9.1 no parece conveniente: por un lado tenemos que «ir y volver» y por otro lado tenemos que conservar los coeficientes q_i.

Observemos que de (9.4),

$$s_{i+1} = t_i = s_{i-1} + q_i s_i, \tag{9.5}$$

por lo que podríamos ir «hacia adelante» sin necesidad de conservar los valores de q_i, sólo necesitamos conservar los dos últimos valores de s_i. El problema es que no conocemos valores iniciales de s_i.

Esto sugiere que una forma de mejorar el procedimiento, haciendo sólo una «pasada hacia adelante», es escribir los restos sucesivos en términos de a y b, encontrando u_i y v_i tales que

$$r_i = u_i a + v_i b \quad \text{para } i \geq 1. \tag{9.6}$$

Recordando (9.1), ponemos inicialmente

$$\begin{aligned} r_{-1} = a, \quad u_{-1} = 1, \quad v_{-1} = 0, \\ r_0 = b, \quad u_0 = 0, \quad v_0 = 1, \end{aligned} \tag{9.7}$$

y si para $j \geq 1$ tenemos

$$r_i = u_i a + v_i b \quad \text{cuando } i = j-1, j-2,$$

de (8.1) quedará

$$\begin{aligned} r_j &= u_{j-2} a + v_{j-2} b - q_j \left(u_{j-1} a + v_{j-1} b \right) \\ &= \left(u_{j-2} - q_j u_{j-1} \right) a + \left(v_{j-2} - q_i v_{j-1} \right) b, \end{aligned}$$

por lo que

$$u_j = u_{j-2} - q_j u_{j-1} \quad \text{para } j \geq 1, \tag{9.8}$$

y una relación similar para v_j.

E 9.2 (extensión del algoritmo de Euclides I). Siguiendo con las notaciones anteriores, supongamos que a y b son enteros, $d = \text{mcd}(a, b)$, $a' = a/d$, $b' = b/d$, y $r_k = d$ es el último resto no nulo con $k \geq 1$.

a) a y b son no nulos y $d < \text{mín}\,\{|a|, |b|\}$.

b) Si $0 \le j \le k+1$ entonces $u_j v_{j-1} - v_j u_{j-1} = (-1)^j$.

c) u_j y v_j son coprimos para $1 \le j \le k+1$.

d) $|u_{k+1}| = |b'|$ y $|v_{k+1}| = |a'|$.

e) $|u_k| \le |b'|/2$ y $|v_k| \le |a'|/2$.

Sugerencia: Hacer inducción en k.

✍ Ver el ejercicio 9.5. ✂

E 9.3 (extensión del algoritmo de Euclides II). Para plasmar las condiciones anteriores en un algoritmo para escribir $r_k = \text{mcd}(a,b)$ como

$$\text{mcd}(a,b) = ua + vb,$$

observemos que el conocimiento de u (y de r_k) implica el conocimiento de v, lo que nos lleva a un esquema como:

```
d = abs(a)
r = abs(b)
uvie = 1    # más viejo
unue = 0    # más nuevo

while r != 0:
    q, d, r = d // r, r, d % r
    unue, uvie = uvie - q * unue, unue

# acá r == 0, d = mcd(abs(a), abs(b))

# no nos interesa u nuevo, sino u viejo
if a < 0:
    uvie = -uvie

# calculamos v a partir de d = u a + v abs(b)
v = (d - uvie * a) // b
```

a) Ver que si a y b no se anulan, al finalizar se satisface $d = ua + vb$, efectivamente obteniendo una solución a la identidad de Bézout (6.1) (todos los números involucrados son enteros).

b) ¿Qué sucede cuando alguno de a o b se anula?

c) Definir una función `mcdext` para encontrar los coeficientes u y v, completando los pasos que sean necesarios.

✍ No es claro quién presentó originalmente la extensión del algoritmo de Euclides para obtener la identidad de Bézout.

Según Knuth (1998), hay indicios en el tratado de astronomía *Aryabhatiya* (año 499 d. C.) escrito en sánscrito por el hindú Aryabhata.

El «truco» de la representación (9.7) se atribuye a W. A. Blankinship (1963), quien lo presentó en una forma matricial para varias incógnitas. Por otra parte, la simplificación que evita el cálculo explícito de los sucesivos v_i, parece deberse a G. H. Bradley (1970).

✂

E 9.4 (ecuaciones diofánticas lineales de dos incógnitas). Queremos encontrar soluciones enteras de la ecuación

$$ax + by = c, \tag{9.9}$$

donde a, b y c son enteros.

✍ Si x y y fueran arbitrarios, (9.9) define una recta en el plano, y se trata de encontrar los puntos de la recta que tienen ambas coordenadas enteras, como ilustramos en la figura 9.1. A partir de esa figura podemos sospechar que es posible que no haya soluciones. Por otra parte, parece claro que si una recta arbitraria pasa por dos puntos distintos con ambas coordenadas enteras, entonces la pendiente es racional y la recta pasará por infinitos puntos del retículo (ya sea por propiedades del mcd o del mcm). Escarbando un poco más, como la pendiente de la recta definida en (9.9) es racional, si la recta toca a un punto de coordenadas enteras, entonces pasará por infinitos puntos del retículo.

En lo que sigue, ponemos $d = \mathrm{mcd}(a, b)$.

a) Si $a = b = 0$ en la ecuación (9.9), entonces si $c = 0$ cualquier par (x, y) es solución, mientras que si $c \neq 0$ no existen soluciones.

b) Si $b = 0$ y $a \neq 0$, entonces (9.9) tiene soluciones si y sólo si $a \mid c$. En ese caso, (x, y) es solución si y sólo si $x = c/a$, con y entero cualquiera.

Un resultado similar vale en el caso $a = 0$ y $b \neq 0$.

c) Si a y b no son ambos nulos y existe una solución entera a la ecuación (9.9), entonces $d \mid c$.

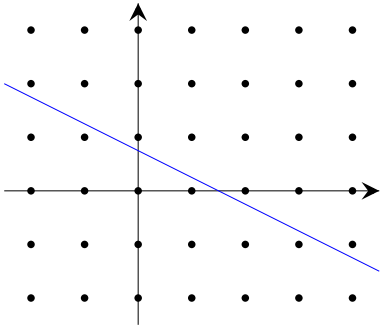


Figura 9.1: Una recta y el retículo de los enteros en el plano.

---

d) Si a y b no son ambos nulos, $a' = a/d$, $b' = b/d$, y (x_0, y_0) es una solución entera de (9.9), entonces:

i) Cualquier otra solución entera (x_1, y_1) es de la forma

$$x_1 = x_0 + kb', \quad y_1 = y_0 - ka' \qquad \text{para algún } k \in \mathbb{Z}. \tag{9.10}$$

ii) Recíprocamente, si $k \in \mathbb{Z}$ y (x_1, y_1) se define por (9.10), entonces (x_1, y_1) es solución entera de (9.9).

e) Usando la identidad de Bézout (6.1), $d = ua + vb$, encontrar todas las soluciones de (9.9) cuando $d \mid c$.

f) A partir de la función encontrada en el ejercicio 9.3, construir una función `diofantica(a, b, c)` retornando los valores x_0, b', y_0, a' de modo que las soluciones de la ecuación (9.9) son de la forma (9.10) (con k arbitrario).

g) (*Gentile*) Encontrar (caracterizar) todas las soluciones enteras de cada una de las siguientes ecuaciones «a mano»:

$$243x + 198y = 9, \quad 71x - 50y = 1,$$

y luego usando la función del apartado anterior.

✍ No es sencillo generalizar el método al caso de más de dos variables. Los interesados pueden consultar el libro de Knuth (1998). ✂

E 9.5 (extensión del algoritmo de Euclides III). Con las notaciones del ejercicio 9.2, ver que si $1 = xa' + yb'$, con x y y enteros, entonces $|u_k| \leq |x|$ y $|v_k| \leq |y|$.

En otras palabras, los coeficientes de la identidad de Bézout obtenidos por el algoritmo de Euclides extendido, son «esencialmente» mínimos.

Ayuda: repasar la ecuación (9.10).

E 9.6. Muchas veces se trata de encontrar las soluciones no negativas de ecuaciones del tipo (9.9) donde a, b y c son positivos.

a) Ver que en este caso hay un número finito de soluciones (podría no haberlas).

b) Definir una función para construir una lista de pares (x, y) de todas las soluciones en este caso (la lista será vacía si no hay soluciones) y aplicarla en los siguientes ejemplos.

✍ Algunos de los siguientes están tomados del libro de *Olds*, y otros del de *Gentile*.

i) (Olds) Un granjero compró un cierto número de vacas a 80 cada una, y un cierto número de cerdos a 50 cada uno. En total pagó 810. ¿Cuántas vacas y cuántos cerdos compró?

ii) (Gentile) En una granja hay conejos y gallinas. Se cuentan 75 cabezas y 244 patas. Determinar cuántos conejos y gallinas hay.

✍ ¡No hay rengos! ☺

iii) (Gentile) En un cine cobran 18 pesos a mayores, y 7.50 pesos a los menores. En un cierto día se recaudaron 900 pesos y asistieron más adultos que menores. ¿Cuáles fueron los posibles números de asistentes?

iv) (Gentile) Dos productos A y B cuestan 7.10 y 8.30 pesos por unidad, respectivamente, ¿qué cantidades enteras de ambos pueden comprarse con 167 pesos?

v) (Olds, Gentile) Cinco marineros náufragos llegaron a la costa de una isla. Para alimentarse juntaron todos los cocos que pudieron encontrar.

Durante la noche, uno de los marineros se despertó y decidió tomar su parte de los cocos. Separó los cocos en 5 pilas y viendo

que sobraba un coco, se lo entregó a los monos. Escondió su parte de cocos y volvió a dormir.

Algo más tarde, un segundo marinero se despertó y tuvo la misma idea que el primero: dividió las pilas en 5 partes iguales, vio que sobraba un coco que entregó a los monos, y después de esconder su parte se fue a dormir.

Lo mismo sucedió con los marineros restantes. Cada uno dividió los cocos que quedaban en cinco pilas, arrojó uno que sobraba a los monos, escondió su parte, y volvió a dormir.

A la mañana siguiente, cada marinero trató de parecer lo más inocente que pudo. Con los cocos restantes hicieron 5 pilas, pero esta vez no sobró ningún coco.

¿Cuál es el mínimo número de cocos que podía tener originalmente la pila de cocos?

E 9.7 (ecuaciones lineales en congruencias). Si a, b y c son enteros, $c > 1$, la ecuación

$$ax \equiv b \pmod{c}, \tag{9.11}$$

se reduce a la ecuación

$$b = ax + cy,$$

donde no interesa el valor de y.

En los siguientes apartados, pediremos $0 \le x < c$.

a) Dar condiciones necesarias y suficientes sobre a, b y c para que exista una solución. ¿Y si se pide unicidad?

b) Determinar la cantidad de soluciones cuando $\mcd(a,c) > 1$ y $\mcd(a,c) \mid b$.

c) Tomando como base la función construida en el ejercicio 9.3.*f*), definir una función para resolver la ecuación (9.11), donde la solución x (cuando existe) satisface $0 \le x < c$.

> Con las notaciones del ejercicio 9.3, bastará encontrar el resto de la división de x_0 por c.

d) (*Gentile*) Usando la función del apartado anterior, resolver las siguientes ecuaciones:

i) $2x \equiv -21 \pmod{8}$.

ii) $2x \equiv -12 \pmod{7}$.

iii) $3x \equiv 5 \pmod 4$.

iv) $10x \equiv 15 \pmod{35}$.

v) $25x \equiv 15 \pmod{29}$.

vi) $36x \equiv 18 \pmod{102}$.

E 9.8 («pequeño» teorema de Fermat). Sean $a \in \mathbb{Z}$ y p primo tales que $p \nmid a$.

a) Sea $A = \{k : 1 \le k \le p-1\}$, y definamos y $f : A \to A$ por

$$f(k) = \text{resto}(a \times k, p) \quad \text{para } k \in A.$$

Ver que f es inyectiva, y por lo tanto biyectiva.
Sugerencia: ver los ejercicios 6.5 y 2.9.

b) $(p-1)! \underset{p}{\equiv} (a \times 1) \times (a \times 2) \times \cdots \times (a \times (p-1)) = a^{p-1} \times (p-1)!$

c) Demostrar el *«pequeño» teorema de Fermat*:

Si p es primo, $a \in \mathbb{Z}$ y $p \nmid a$, entonces

$$a^{p-1} \equiv 1 \pmod p.$$

Por lo tanto, $a^p \equiv a \pmod p$ para todo $a \in \mathbb{Z}$.

✍ Hay demostraciones que usan otras ideas. Por ejemplo a partir de la identidad

$$(a+1)^p = \sum_{j=0}^{p} \binom{p}{j} a^j,$$

y usando que $\binom{p}{j}$ es múltiplo de p para $1 \le j \le p-1$, se obtiene

$$(a+1)^p \underset{p}{\equiv} a^p + 1,$$

y por inducción, que $a^p \underset{p}{\equiv} a$ para todo a (y entonces $a^{p-1} \underset{p}{\equiv} 1$ si $p \nmid a$).

E 9.9. Si a es un entero no nulo y p es un primo tal que $p \nmid a$, entonces $b = \text{resto}\left(a^{p-2}, p\right)$ satisface $a \times b \equiv 1 \pmod p$.
Además, $a = b$ si y sólo si $a = 1$ o $a = p-1$.

E 9.10 (*Gentile*). Sean $n \in \mathbb{N}$, p primo y $m \in \mathbb{N}$ tales que $p^m \ge n$. Entonces la máxima potencia p^k que divide a $n!$ satisface

$$k = \sum_{i=1}^{m} \left\lfloor \frac{n}{p^i} \right\rfloor$$

E 9.11 (teorema de Wilson). Sea n un entero mayor que 1.

a) Si n es primo entonces $(n-1)! \underset{n}{\equiv} -1$.

Sugerencia: usar el ejercicio 9.9 si n es impar.

b) Si $n \neq 4$ y no es primo, entonces $(n-1)! \underset{n}{\equiv} 0$.

Sugerencia: factorizar en primos y usar el ejercicio 9.10.

c) Si $(n-1)! \underset{n}{\equiv} -1$ si y sólo si n es primo.

d) A pesar de la caracterización del apartado anterior, este resultado no se usa computacionalmente para verificar si un número es primo. ¿Por qué?

Atención: El problema no es el tamaño de $(n-1)!$ pues siempre se puede trabajar con números menores que n^2 (tomando restos).

Sugerencia: Comparar la cantidad de bits de n y la cantidad de multiplicaciones para calcular $(n-1)!$.

✍ Waring (1736-1798) publicó —sin demostración— que si p es primo entonces divide a $1+(p-1)!$ y atribuyó el resultado a Wilson (1741–1793), posiblemente ninguno de ellos supiera cómo demostrarlo. Abu Ali al-Hasan ibn al-Haytham (965–1040) ya conocía el resultado:

> *Encontrar un número que dividido por* 2, 3, 4, 5 *y* 6 *da resto* 1 *y es múltiplo de* 7.

El resultado fue probado formalmente por Lagrange en 1773, quien demostró también la recíproca. (Fuente: `http://www-history.mcs.st-andrews.ac.uk/Biographies/Al-Haytham.html`). ✂

E 9.12 (potencias en congruencias). Supongamos que a, b y c son enteros positivos, $c > 1$.

a) Definir una función `potmod(a, b, c)` para encontrar el resto de la división de a^b por c modificando la función definida en el ejercicio 4.4.*c*) comenzando con el resto de la división de a por c y tomando los restos de las divisiones por c en cada paso.

✍ Recordemos que Python tiene la función `pow(a, b, c)`.

b) (*Gentile*) Resolver los siguientes problemas «a mano» y con la función del apartado anterior, viendo que los resultados son idénticos.

✍ Recordar el *pequeño teorema de Fermat* (ejercicio 9.8).

i) Hallar el resto de la división por 7 de 9999^{9999}.

ii) Calcular $7077^{377} \pmod{11}$.

iii) Calcular $100^{150} \pmod{7}$.

iv) Calcular los restos de la división de 3^{999} por 7, 5, 61, 17, 9 y 15.

v) Hallar la última cifra de 2^{100} en su expresión decimal.

vi) Hallar las dos últimas cifras de la expresión decimal de 7^{7^7}.

vii) Hallar x tal que $5^{50} \equiv x \pmod{30}$, $0 \le x < 30$. ✂

E 9.13 (inversos multiplicativos). Un caso particularmente interesante de ecuaciones en congruencias es encontrar el inverso multiplicativo de $a \pmod{b}$, concretamente, resolver la ecuación

$$ax \equiv 1 \pmod{b},$$

donde a y b son enteros, $b > 1$ y $\mathrm{mcd}(a,b) = 1$.

Definir una función `invmod(a, b)` para resolver este problema, que no siempre tiene solución. ✂

E 9.14 (teorema chino del resto). En este ejercicio vemos una demostración pautada del teorema que establece:

> *Si $m_1, m_2, \ldots, m_k$ son enteros positivos coprimos dos a dos, es decir,* $\mathrm{mcd}(m_i, m_j) = 1$ *para* $i \neq j$, *y* $m = m_1 \times m_2 \times \cdots \times m_k$, *entonces, para cualesquiera enteros* $r_1, r_2, \ldots, r_k$ *existe un único* x, $0 \le x < m$, *tal que*
>
> $$x \equiv r_i \pmod{m_i} \quad \textit{para } i = 1,\ldots,k. \tag{9.12}$$

✍ Para los que sepan álgebra lineal: podemos pensar que la demostración se reduce a estudiar «el caso de la base canónica», y luego usar linealidad. Es decir, un primer paso sería encontrar x_j tal que

$$x_j \equiv \begin{cases} 1 \pmod{m_j} \\ 0 \pmod{m_i} \quad \text{para } j \neq i, \end{cases} \tag{9.13a}$$

y luego tomar

$$x = \sum_i r_i x_i. \tag{9.13b}$$

La siguiente demostración considera primero el caso $k = 2$ y luego usa inducción, que en la computadora se traduce como repetición o recursión.

a) Demostrar el resultado para $k = 2$ tomando α_1 y α_2 tales que

$$\alpha_1 m_1 + \alpha_2 m_2 = 1, \tag{9.14a}$$

y luego

$$x \equiv r_1 \alpha_2 m_2 + r_2 \alpha_1 m_1 \pmod{m_1 m_2}. \tag{9.14b}$$

✍ Acordarse de demostrar también unicidad, no sólo existencia.

✍ Pensando en las ecuaciones (9.13), la ecuación (9.14a) expresa a (9.13a), pues

$$\alpha_2 m_2 \equiv \begin{cases} 1 \pmod{m_1} \\ 0 \pmod{m_2} \end{cases}, \quad \alpha_1 m_1 \equiv \begin{cases} 1 \pmod{m_2} \\ 0 \pmod{m_1} \end{cases},$$

con lo que (9.14b) es un reflejo de (9.13b) cuando $k = 2$.

b) Demostrar el resultado para k cualquiera usando inducción. ✂

E 9.15. Definir una función para implementar el teorema chino del resto, donde la entrada son los números $m_1, m_2, \ldots, m_k$ y $r_1, r_2, \ldots, r_k$ (ya sea como dos listas separadas cada una de longitud k, o como k pares de la forma $(m_1, r_1), (m_2, r_2), \ldots, (m_k, r_k)$).

La función podría tener un lazo donde, después de trabajar con m_1 y m_2 se van incorporando de a uno los módulos $m_3, m_4, \ldots$ ✂

E 9.16 (Sun Tzu). El nombre de teorema *chino* del resto se debe a que una de las primeras citas del resultado que se conocen es la que aparece en el libro «Sunzi Suanjing»: *El clásico matemático de Sunzi*, escrito por Sunzi (Sun Tzu) (entre siglo III y V):

> *Hay ciertas cosas cuya cantidad se desconoce.*
> *Cuando divididas por 3 tienen resto 2;*
> *cuando divididas por 5 tienen resto 3;*
> *y cuando divididas por 7 el resto es 2.*
> *¿Cuál será la cantidad?*

Resolver este acertijo.

✍ Enunciados similares al de Sun Tzu aparecieron en muchas partes, por ejemplo en el *Liber Abaci* de Fibonacci (1202). *Gentile* tiene un ejercicio con los mismos números.

✍ Otro Sun Tzu, seguramente más famoso, fue el autor del *Arte de la guerra* (544–496 a. C.). ✂

E 9.17 (el adivino). Se pide a un amigo que piense un número entre 1 y 100 y diga cuáles son los restos de ese número cuando divido por 3, 5 y 7.

✍ Observar la similitud con el enunciado de Sun Tzu. ✂

E 9.18. Resolver los siguientes ejercicios tomados de *Gentile*, usando la función construida en el ejercicio 9.15.

a) En un campo hay un cierto número de cabezas de ganado. Contados de a 4, de a 6 y de a 15 sobran 2, mientras que contados de a 7 no sobra ninguno. ¿Cuál es el mínimo número de cabezas que hay?

b) Hallar un número tal que dividido por 2, 3, 6 y 12 de restos 1, 2, 5 y 5 respectivamente.

c) Hallar el menor entero $a > 1$ tal que verifique simultáneamente

$$a \equiv 2 \pmod 4, \quad a \equiv 1 \pmod 5, \quad a \equiv 1 \pmod 7.$$

d) Hallar 4 enteros consecutivos divisibles respectivamente por 5, 7, 9 y 11.

e) En una granja la producción diaria de huevos es inferior a 75. Cierto día el recolector informó que la cantidad recogida era tal que contando de a 3 sobraban 2, contando de a 5 sobraban 4, y contados de a 7 sobraban 5. El capataz dijo que era imposible. ¿Quién tenía razón?

f) Hallar el menor $x > 0$ que satisfaga simultáneamente

$$3x \equiv 1 \pmod 8, \quad 5x \equiv 3 \pmod 7, \quad 4x \equiv 2 \pmod 5.$$

g) Un niño recibe una cierta cantidad de dinero de su padre. Este dinero lo puede recibir de tres maneras diferentes:

i) Anticipo de 1 peso y el resto en cuotas mensuales de 39 pesos.

ii) Anticipo de 10 pesos y el resto en cuotas mensuales de 11 pesos.

iii) Anticipo de 2 pesos y el resto en cuotas mensuales de 7 pesos.

Determine el monto mínimo a recibir, y la cantidad de cuotas mensuales respectivas. ✂

E 9.19 (función ϕ de Euler). Dado $n \in \mathbb{N}$, consideremos el conjunto

$$\mathscr{M}(n) = \{z \in \mathbb{Z} : 1 \le z < n \text{ y } \operatorname{mcd}(n,z) = 1\},$$

y definamos la función ϕ *de Euler* como su cardinal,

$$\phi(n) = \#(\mathscr{M}(n)).$$

a) Probar que n es primo $\Leftrightarrow \phi(n) = n-1$.

b) Supongamos que $\operatorname{mcd}(m,n) = 1$, y consideremos $u = m \times n$ y los conjuntos

$$M = \{a \in \mathbb{N} : 1 \le a \le m, \operatorname{mcd}(a,m) = 1\},$$
$$N = \{a \in \mathbb{N} : 1 \le a \le n, \operatorname{mcd}(a,n) = 1\},$$
$$U = \{a \in \mathbb{N} : 1 \le a \le u, \operatorname{mcd}(a,u) = 1\},$$

y definamos una función $f : U \to M \times N$ poniendo

$$f(c) = (\operatorname{resto}(c,m), \operatorname{resto}(c,n)).$$

Ver que f es biyectiva, y por lo tanto los cardinales de U y $M \times N$ coinciden.

Ayuda: usar el teorema chino del resto (y recordar los ejercicios 2.5 y 2.7).

c) Ver que ϕ es *multiplicativa*, esto es, si m y n son enteros positivos con $\operatorname{mcd}(m,n) = 1$, entonces $\phi(m \times n) = \phi(m) \times \phi(n)$.

d) Si $n = p^\alpha$ para algún primo p y $\alpha \in \mathbb{N}$, entonces $\phi(n) = (p-1)\,p^{\alpha-1}$.

e) Si $n > 1$ se descompone en producto de primos como en (7.2), entonces

$$\phi(n) = (p_1 - 1) \times p_1^{\alpha_1 - 1} \times (p_2 - 1) \times p_2^{\alpha_2 - 1} \times \cdots \times (p_k - 1) \times p_k^{\alpha_k - 1},$$

que a veces se escribe como

$$\phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times \cdots \times \left(1 - \frac{1}{p_k}\right)$$

f) Construir dos funciones para calcular $\phi(n)$ con $n \in \mathbb{N}$: una construyendo la lista de coprimos y calculando su longitud, y la otra usando la descomposición en primos mencionada anteriormente.

- ✍ Tener en cuenta el caso $n = 1$.
- ✍ Una posibilidad para la primera es poner `len([m for m in range(1, n) if mcd(m, n) == 1])`. Para la segunda se podría usar la descomposición en el ejercicio 7.1.*b.ii*).
- ✍ No se conocen algoritmos sencillos (eficientes) para calcular ϕ.

g) Calcular $\phi(n)$ para varios valores de n, y después conjeturar y demostrar para cuáles valores de n, $\phi(n)$ es par.

h) Recordando el resultado de Gauss y Wantzel sobre la posibilidad de construir un polígono regular de n lados ($n \geq 3$) que mencionamos en el ejercicio 7.9, demostrar que $\phi(n)$ es una potencia de 2 si y sólo si n es el producto de una potencia de 2 con primos de Fermat distintos, es decir

$$n = 2^m p_1^{\alpha_1} \dots p_k^{\alpha_k},$$

donde $m \geq 0$, $\alpha_i \in \{0,1\}$ y p_i son primos de Fermat. ✂

E 9.20 (teorema de Fermat-Euler). Sea $n \in \mathbb{N}$ y consideremos $\mathscr{M}(n)$ y $\phi(n)$ como en el ejercicio 9.19.

a) Si $n > 1$ y $\mathrm{mcd}(a,n) = 1$, la función $f : \mathscr{M}(n) \to \mathscr{M}(n)$ definida por

$$f(k) = \mathrm{resto}(k \times a, n) \quad \text{para } k \in \mathscr{M}(n),$$

es biyectiva.

Sugerencia: usar las ideas del ejercicio 9.8.

b) Demostrar el teorema de Euler-Fermat:

Si $n \in \mathbb{N}$, $a \in \mathbb{Z}$ *y* $\mathrm{mcd}(a,n) = 1$, *entonces* $a^{\phi(n)} \equiv 1 \pmod{n}$. ✂

Las n fracciones

$$\frac{1}{n}, \frac{2}{n}, \frac{3}{n}, \dots, \frac{n-1}{n}, \frac{n}{n},$$

llevadas a su mínima expresión son de la forma a/b con $\mathrm{mcd}(a,b) = 1$, $1 \leq a \leq b$ y $b \mid n$, y es de esperar que haya una relación entre n y $\phi(d)$ para los divisores d de n, que estudiamos en el siguiente ejercicio.

E 9.21. Sea $n \in \mathbb{N}$ y consideremos

$$B(b) = \{a : 1 \le a \le b, \mathrm{mcd}(a,b) = 1\} \quad \text{para } b \in \mathbb{N} \text{ tal que } b \mid n,$$
$$A = \{(a,b) : 1 \le a \le b \le n, \mathrm{mcd}(a,b) = 1, b \mid n\} = \{(a,b) : b \mid n, a \in B(b)\}.$$

a) $\#(A) = \sum_{b:b|n} \#(B(b)) = \sum_{b:b|n} \phi(b)$.

b) Sea $\mathbb{I}_n = \{1,2,\ldots,n\}$ y consideremos $f : A \to \mathbb{I}_n$ dada por[(5)]

$$f(a,b) = a \times \frac{n}{b}.$$

Ver que f es biyectiva.

c) $n = \sum_{d:d|n} \phi(d)$. ✂

E 9.22. El conjunto de divisores del número natural n,

$$D(n) = \{d \in \mathbb{N} : d \mid n\},$$

que estudiamos en el ejercicio 6.2, da lugar a varias funciones aritméticas: para $s \in \mathbb{R}$ definimos

$$\sigma_s(n) = \sum_{d \in D(n)} d^s,$$

poniendo simplemente σ para indicar σ_1.

Así, para $n = 12$ tenemos

$$D(12) = \{1,2,3,4,6,12\}, \quad \sigma_0(12) = 6, \quad \sigma(12) = \sigma_1(12) = 28.$$

a) Ver que σ_s es multiplicativa para cualquier s, es decir, si a y b son coprimos entonces $\sigma_s(ab) = \sigma_s(a)\sigma_s(b)$.

b) n es primo $\Leftrightarrow \sigma_0(n) = 2 \Leftrightarrow \sigma(n) = n+1$.

c) n es un cuadrado perfecto $\Leftrightarrow \sigma_0(n)$ es impar.

d) Si p es primo y $m \in \mathbb{N}$,

$$\sigma\left(p^m\right) = \frac{p^{m+1}-1}{p-1}.$$

En particular, si $p = 2$ entonces $\sigma(2^m) = 2^{m+1} - 1$. ✂

(5) Se trata de la fracción a/b considerada en el argumento «informal» anterior, multiplicada por n.

E 9.23 (números perfectos). $n \in \mathbb{N}$ es *perfecto* si es la suma de sus divisores positivos propios (o sea, excluyendo n). Equivalentemente, recordando el ejercicio 9.22, n es perfecto si

$$\sigma(n) = 2n.$$

Por ejemplo, 6 es perfecto pues $6 = 1 + 2 + 3$.

a) Encontrar los 4 primeros números perfectos a partir de la suma de sus divisores.
Respuesta: $6, 28, 496, 8128$.

b) Si n es perfecto, entonces

$$\sum_{d:d|n} \frac{1}{d} = 2.$$

¿Es cierta la recíproca, es decir si la suma de los inversos multiplicativos de los divisores es 2 entonces n es perfecto?

En los apartados siguientes vemos que hay una relación biunívoca entre los números perfectos pares y los primos de Mersenne (ejercicio 7.10).

c) Sea p un número primo tal que $2^p - 1$ es primo (de Mersenne). Entonces $n = 2^{p-1}(2^p - 1)$ es perfecto.
Ayuda: calcular $\sigma(n)$, la suma de los divisores de n.

d) Sea n un número perfecto par y escribamos $n = 2^r q$, con q impar.

i) $2n = \left(2^{r+1} - 1\right)\sigma(q)$.

ii) Existe $m \in \mathbb{N}$ tal que $\sigma(q) = m2^{r+1}$ y $q = m\left(2^{r+1} - 1\right)$, por lo tanto

$$m \mid q \quad \text{y} \quad \sigma(q) = q + m.$$

iii) $q = 2^{r+1} - 1$ es primo (de Mersenne).

e) Por lo tanto n es un número perfecto para $\Leftrightarrow$ $n = 2^{p-1}(2^p - 1)$ donde $2^p - 1$ es primo (y por lo tanto también lo es p).

✍ No se sabe si hay números perfectos impares.

✍ Euclides da la definición de número perfecto en VII.22, y en IX.36 demuestra *c*).
Euler demostró la recíproca, publicada póstumamente en 1849. La demostración en *d*) se debe a L. E. Dickson (1911), y es la que presenta *Gentile*.

10. Fracciones continuas

En esta sección hacemos una introducción a las *fracciones continuas*, que en cierta forma extienden el algoritmo de Euclides a números irracionales, y repetiremos algunas ideas de la sección 9.

Además de ser atrayentes en sí mismas, las fracciones continuas tienen aplicaciones teóricas como la aproximación numérica de irracionales mediante racionales o construcción de soluciones enteras de la ecuación $x^2+y^2=p$ cuando p es primo y $p \underset{4}{\equiv} 1$, o prácticas como factorización en primos, temas que dejaremos para otra oportunidad.

En la sección 10.1 veremos que así como los racionales tienen expresiones decimales periódicas, las fracciones continuas periódicas están asociadas a las raíces irracionales de ecuaciones cuadráticas con coeficientes enteros. Esto permite resolver algunas ecuaciones diofánticas como $x^2-my^2=\pm 1$, llamada *ecuación de Pell*, el primer paso en el estudio de formas cuadráticas enteras.

La bibliografía básica para esta sección son los libros de Olds (1963) (más elemental) y de Hardy y Wright (2008) (más completo), aunque las notaciones acá pueden diferir un poco.

✍ *Fracciones continuas* parece una mala traducción del inglés *continued fractions*, tal vez sería mejor «fracciones continuadas». Como en el caso de «grafo», «máximo común divisor» o «fuerza conservativa» en física, lo prudencia aconseja usar la nomenclatura establecida para no introducir nuevos términos que lleven a confusión.

Una *fracción continua* es una expresión de la forma

$$a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\ddots}}, \tag{10.1}$$

donde $a_0, a_1, \ldots$ son números reales, $a_i>0$ para $i\geq 1$, y normalmente denotamos la expresión (10.1) más simplemente por

$$[a_0, a_1, \ldots]. \tag{10.2}$$

Observar que la condición $a_i>0$ para $i\geq 1$ garantiza que nunca dividiremos por 0.

✍ Hay fracciones continuas más generales, donde los numeradores no son necesariamente 1 o los denominadores necesariamente positivos. En algunos contextos, las que vemos acá se llaman *simples*.

Por ejemplo, $[3,4,5]$ corresponde al número

$$3+\cfrac{1}{4+\cfrac{1}{5}}=3+\frac{1}{21/5}=\frac{3\times 21+5}{21}=\frac{68}{21}=3.\overline{238095}.$$

Recíprocamente, para encontrar una expresión como fracción continua de 68/21 podemos poner

$$\frac{68}{21}=3+\frac{5}{21},\quad \frac{21}{5}=4+\frac{1}{5},\quad \frac{5}{1}=5,$$

que no es sino seguir los pasos del algoritmo de Euclides con restos: los cocientes sucesivos son los coeficientes que aparecen en la fracción continua.

Una fracción continua es *finita* si tiene un número finito de términos, o sea, es de la forma $[a_0,a_1,\ldots,a_n]$. Por otro lado, diremos que la fracción continua es *entera* si todos los coeficientes $a_0,a_1,\ldots$ son enteros.

Así, la fracción continua $[3,4,5]$ que acabamos de ver es finita y entera.

Dada la fracción continua $[a_0,a_1,\ldots]$ (finita o no, entera o no), el *n-ésimo convergente* es la n-ésima fracción parcial

$$c_n=[a_0,a_1,\ldots,a_n],$$

(cuando todos los coeficientes $a_0,a_1,\ldots,a_n$ están definidos), y observamos que

$$c_0=[a_0]=\frac{a_0}{1}, \tag{10.3}$$

$$c_1=[a_0,a_1]=a_0+\frac{1}{a_1}=\frac{a_0a_1+1}{a_1}, \tag{10.4}$$

$$c_n=[a_0,a_1,\ldots,a_n]=\left[a_0,a_1,\ldots,a_{n-2},a_{n-1}+\frac{1}{a_n}\right], \tag{10.5}$$

$$[a_0,a_1,\ldots,a_n]=a_0+\frac{1}{[a_1,a_2,\ldots,a_n]}, \tag{10.6}$$

de modo que (10.3) y (10.4) junto con (10.5) o (10.6) permiten dar una definición inductiva de la expresiones (10.1) y (10.2). Más aún, para valores adecuados de m y n, (10.5) y (10.6) pueden generalizarse a

$$\begin{aligned}[a_0,a_1,\ldots,a_n] &= [a_0,a_1,\ldots,a_{m-1},[a_m,\ldots,a_n]] \\ &= \left[a_0,a_1,\ldots,a_{m-2},a_{m-1}+\frac{1}{[a_m,\ldots,a_n]}\right] \end{aligned} \tag{10.7}$$

> *Para no estar poniendo en cada momento dónde termina una fracción continua finita, en lo sucesivo entenderemos que expresiones que involucren términos de la forma a_k sólo se enuncian para $k \le n$ si la fracción continua es $[a_0,a_1,\ldots,a_n]$.*

E 10.1. Ver que si $[a_0,a_1,\ldots,a_n]$ es una fracción continua entera finita, entonces $a_0 = \lfloor [a_0,a_1,\ldots,a_n] \rfloor$, donde $\lfloor x \rfloor = \max\{z \in \mathbb{Z} : z \le x\}$ es la función piso.

Sugerencia: usar (10.6). ✂

E 10.2. *a*) Toda fracción continua finita entera representa un número racional.

b) Todo número racional positivo puede escribirse como fracción continua finita entera de exactamente dos formas: una terminando en un entero $z > 1$, y la otra con un elemento más terminando en $z-1$ y 1.

c) ¿Y si el racional fuera negativo? ✂

E 10.3. Resolver los siguientes tomando como base el ejemplo de $[3,4,5]$ y 68/21.

✍ Recordar la nota del ejercicio 4.3.*c*).

a) Construir una función `fcar` (fracción continua a racional) para encontrar numerador y denominador del racional representado por la fracción continua finita entera $[a_0,a_1,\ldots,a_n]$.

Por ejemplo `fcar([3, 4, 5])` → `[68, 21]`.

b) Recíprocamente, construir una función `rafc` (racional a fracción continua) para encontrar una fracción continua finita entera asociada a un racional dados numerador y denominador.

Por ejemplo `rafc(-68, 21)` → `[-4, 1, 3, 5]` (comprobar también el comportamiento de `rafc(68, -21)`). ✂

Como sabemos, los números reales tienen expresiones decimales «infinitas», y podemos distinguir a los racionales porque sus expresiones terminan siendo periódicas. Más aún, algunos números racionales tienen dos: una con período 9 y otra con período 0, como $1.23 = 1.23\overline{0} = 1.22\overline{9}$, siendo exactamente los racionales no nulos que pueden escribirse como $m/10^n$ con $m \in \mathbb{Z}$ y $n \in \mathbb{N}$.

Al pasar a fracciones continuas no queda claro qué sería una fracción continua infinita. Desde el punto de vista formal, no hay inconvenientes en considerar una expresión de la forma (10.2) como $[1,2,3,4,\ldots]$, pero nos gustaría saber si representa a algún número, y en ese caso cuál, ya que no es posible evaluarla «viniendo desde atrás» como hicimos con $[3,4,5]$.

El caso más sencillo de fracción continua infinita es $[1,1,1,\ldots]$, es decir $a_n = 1$ para todo $n \geq 0$. Si tomamos los primeros convergentes obtenemos:

$$c_0 = [1] = \frac{1}{1} = 1, \quad c_1 = [1,1] = \frac{2}{1} = 2, \quad c_2 = [1,1,1] = \frac{3}{2} = 1.5,$$
$$c_3 = [1,1,1,1] = \frac{5}{3} = 1.666\ldots, \quad c_4 = [1,1,1,1,1] = \frac{8}{5} = 1.6,$$
$$c_5 = [1,1,1,1,1,1] = \frac{13}{8} = 1.625,$$

y observamos dos cosas:

- En este caso particular, vamos obteniendo los cocientes entre números de Fibonacci sucesivos que, como sabemos,[6] se aproximan al número de oro $\tau = \left(1+\sqrt{5}\right)/2$.
- Los valores de los convergentes con subíndice par van creciendo, los correspondientes a subíndices impares van decreciendo y son siempre mayores que los que tienen subíndices pares:

[6] Recordar la fórmula de Binet.

$$\frac{1}{1} = 1 < \frac{3}{2} = 1.5 < \frac{8}{5} = 1.6 < \cdots$$
$$< \frac{13}{8} = 1.625 < \frac{5}{3} = 1.666\cdots < \frac{2}{1} = 2,$$

lo que —como veremos— es un hecho general.

E 10.4. Con las notaciones anteriores, queremos construir p_n y q_n de modo que $c_n = p_n/q_n$ para $n \geq 0$.

a) Inicialmente podemos poner

$$p_0 = a_0, \quad q_0 = 1, \quad p_1 = a_0a_1 + 1, \quad q_1 = a_1, \tag{10.8a}$$

b) Usando inducción, ver que para $n \geq 2$, podemos poner

$$p_n = a_n p_{n-1} + p_{n-2}, \quad q_n = a_n q_{n-1} + q_{n-2}. \tag{10.8b}$$

Sugerencia: poner $[\tilde{a}_0, \tilde{a}_1, \ldots, \tilde{a}_{n-1}]$, donde $\tilde{a}_k = a_k$ para $k < n-1$ y $\tilde{a}_{n-1} = a_{n-1} + 1/a_n$, considerar los convergentes $\tilde{p}_k/\tilde{q}_k$ y usar (10.5).

✍ El trabajo que estamos haciendo es una variante de lo que hicimos en la sección 9.

Para calcular los coeficientes u y v de la identidad de Bézout, en el ejercicio 9.1 hicimos una primera versión donde «vamos hacia adelante y volvemos para atrás», similar a lo que hicimos en el ejercicio 10.3 al recorrer la lista empezando «desde atrás».

Luego vimos una versión «yendo sólo para adelante», equivalente al estudio de convergentes. Por ejemplo, comparar los valores iniciales de p_0 y q_0 en (10.8a) con los valores iniciales en (9.7), y las definiciones de los convergentes en (10.8b) con los valores en (9.5) y (9.8). ✂

E 10.5. Rehacer el programa en el ejercicio 10.3.*a*) usando las ecuaciones (10.8) (y yendo de «adelante para atrás»). ✂

E 10.6. Sea $[a_0, a_1, \ldots]$ una fracción continua entera (finita o no), y definamos p_n y q_n usando las ecuaciones (10.8).

a) $q_n > 0$ y $q_{n+2} > a_{n+2}q_{n+1}$ para todo n.

b) Si $a_0 \geq 0$ entonces $p_n > 0$ y $p_{n+2} > a_{n+2}p_{n+1}$ para todo n.

c) Para $n \geq 0$,

$$p_{n+1}q_n - p_n q_{n+1} = (-1)^n,$$

es decir,

$$c_{n+1} - c_n = (-1)^n/(q_n q_{n+1}).$$

Ayuda: $p_1 q_0 - p_0 q_1 = 1$.

d) Si $n \geq 0$, $p_{n+2}q_n - p_n q_{n+2} = (-1)^n a_{n+2}$, y por lo tanto

$$c_{n+2} - c_n = (-1)^{n+1} a_{n+2}/(q_n q_{n+2}).$$

e) $c_n < c_{n+2}$ si n es par y $c_{n+2} < c_n$ si n es impar.

f) $c_k < c_j$ si k es par y j impar.

✍ Podemos resumir los dos últimos apartados poniendo informalmente

$$c_0 < c_2 < c_4 < \cdots < c_5 < c_3 < c_1.$$

g) $\mathrm{mcd}(p_n, q_n) = 1$ para todo $n \geq 0$, y por lo tanto la fracción p_n/q_n está en su «mínima expresión». ✂

E 10.7. Con las notaciones anteriores,

$$[a_0, a_1, \ldots, a_n] = \frac{[a_m, \ldots, a_n]\, p_{m-1} + p_{m-2}}{[a_m, \ldots, a_n]\, q_{m-1} + q_{m-2}} \quad \text{para } 0 < m < n. \qquad (10.9)$$

Ayuda: usar (10.7). ✂

E 10.8 (series de Farey). Para $n \in \mathbb{N}$, la *serie de Farey de orden* n, $\mathfrak{F}_n$, es el conjunto de fracciones irreducibles de la forma a/b, con $1 \leq b \leq n$ y $0 \leq a \leq b$, con las convenciones $0 = 0/1$ and $1 = 1/1$. Por ejemplo, $\mathfrak{F}_4 = (0, 1/4, 1/3, 1/2, 2/3, 3/4, 1)$.

a) ¿Qué relación hay entre $\mathfrak{F}_n$ y los conjuntos A y B considerados en el ejercicio 9.21?, ¿y entre la función ϕ de Euler y $\#(\mathfrak{F}_n)$?

b) Si h/k y h'/k' son dos términos sucesivos de $\mathfrak{F}_n$, entonces

$$kh' - hk' = 1.$$

c) Si h/k, h''/k'' y h'/k' son tres términos sucesivos de $\mathfrak{F}_n$, entonces

$$\frac{h''}{k''} = \frac{h + h'}{k + k'}.$$

d) El promedio en *c*) puede ser usado para demostrar que hay infinitos racionales: si a, b, c y d son naturales y $a/b < c/d$, entonces $(a+b)/(c+d)$ es racional y

$$\frac{a}{b} < \frac{a+c}{b+d} < \frac{c}{d}.$$

e) Si $n > 1$, no hay dos términos consecutivos de $\mathfrak{F}_n$ con el mismo denominador.

f) $a/n \in \mathfrak{F}_n$ si y sólo si $a = h + h'$, $n = k + k'$ donde h/k y h'/k' son dos términos consecutivos de $\mathfrak{F}_{n-1}$.

g) Definir una función en Python para construir, dado $n \in \mathbb{N}$, la sucesión $\mathfrak{F}_n$ como lista de pares.

Por ejemplo,

```
>>> farey(4)
[(0, 1), (1, 4), (1, 3), (1, 2), (2, 3),
(3, 4), (1, 1)]
```

✍ Los apartados *b*), *c*) y *e*) están tomados de Hardy y Wright (2008, cap. 3). Los dos primeros son equivalentes en el sentido de que uno puede deducirse a partir del otro. ✂

Los apartados *a*), *c*), *e*) y *f*) del ejercicio 10.6 muestran que cualquier fracción continua infinita converge a un número irracional (los denominadores son cada vez mayores). Recíprocamente, todo irracional da lugar a una única fracción continua infinita como vemos a continuación.

E 10.9. Sea α un irracional y pongamos

$$\alpha_0 = \alpha, \quad a_0 = \lfloor \alpha \rfloor, \quad \rho_0 = \alpha - a_0, \tag{10.10a}$$

y mientras $\rho_k \neq 0$, definamos inductivamente

$$\alpha_{k+1} = 1/\rho_k, \quad a_{k+1} = \lfloor \alpha_{k+1} \rfloor, \quad \rho_{k+1} = \alpha_{k+1} - a_{k+1}, \tag{10.10b}$$

es decir,

$$\alpha_k = a_k + \rho_k = a_k + \frac{1}{\alpha_{k+1}}. \tag{10.10c}$$

a) Ver que para todo k, α_k y ρ_k son irracionales. En particular, $\rho_k > 0$ (y nunca se anula).

b) Consideremos los convergentes $c_n = [a_0, a_1, \ldots, a_n] = p_n/q_n$, donde p_n y q_n están definidos por las ecuaciones (10.8).

Entonces

$$c_0 < c_2 < \cdots < \alpha < \cdots < c_3 < c_1,$$

y

$$\left|\alpha - \frac{p_n}{q_n}\right| \leq \frac{1}{q_n q_{n+1}} < \frac{1}{q_n^2}.$$

✍ El resultado implica que —dado cualquier irracional α— hay una infinidad de racionales p/q tales que $|\alpha - p/q| < 1/q^2$. Hurwitz demostró en 1891 que el resultado no se puede mejorar mucho para irracionales arbitrarios: hay una infinidad de fracciones p/q tales que

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{\sqrt{5}q^2},$$

y no se puede poner otra constante mayor que $\sqrt{5}$.

No obstante, la cota puede mejorarse para números *trascendentes*, es decir, números que *no* son raíces de ecuaciones polinómicas con coeficientes racionales.

Como es de sospechar, $\sqrt{5}$ está relacionado con el número de oro, los números de Fibonacci y la fracción continua $[1, 1, 1, \ldots]$.

c) Si las fracciones continuas infinitas $[a_0, a_1, \ldots]$ y $[a'_0, a'_1, \ldots]$ convergen al mismo número, entonces $a_i = a'_i$ para todo $i \geq 0$.

Sugerencia: a_n es la parte entera de $a_n + 1/[a_{n+1}, \ldots]$ y usar inducción. ✂

No veremos más sobre fracciones continuas asociadas a números irracionales en general, y solamente dejamos un resumen de esta parte:

- Las fracciones continuas finitas enteras están asociadas a números racionales.

  Cada racional tiene asociadas (exactamente) dos fracciones continuas finitas enteras.
- En fracciones continuas infinitas enteras, «los convergentes convergen» ☺.
- Hay una correspondencia biunívoca entre números irracionales y fracciones continuas infinitas enteras.

10.1. Irracionales cuadráticos

Así como los números que tienen una expresión decimal periódica son de una clase particular, los números que tienen asociada una fracción continua periódica son especiales: un *irracional cuadrático* es una solución irracional de una ecuación de la forma

$$ax^2 + bx + c = 0, \tag{10.11}$$

donde a, b y c son enteros, $a \neq 0$, y $b^2 - 4ac > 0$.

Si una fracción continua entera es periódica (y por lo tanto infinita), digamos

$$\left[a_0, a_1, \ldots, a_{L-1}, \underbrace{a_L, \ldots, a_{L+k-1}}_{\text{período}}, a_L, \ldots\right], \tag{10.12}$$

y representa al número irracional α, llamemos β a la parte periódica, denotando el período con una barra por arriba (como hicimos en el ejercicio 7.11, sólo que ahora hay valores separados por comas):

$$\beta = [a_L, \ldots, a_{L+k-1}, a_L, \ldots] = [\overline{a_L, \ldots, a_{L+k-1}}]. \tag{10.13}$$

A partir de las versiones infinitas de (10.7) y (10.9),

$$\beta = \left[a_L, \ldots, a_{L+k-1}, \beta\right] = \frac{p'\beta + p''}{q'\beta + q''}, \tag{10.14}$$

donde p', p'', q' y q'' son enteros definidos mediante las ecuaciones (10.8) y las condiciones $p'/q' = [a_L, \ldots, a_{L+k-1}]$ y $p''/q'' = [a_L, \ldots, a_{L+k-2}]$, por lo que

$$q'\beta^2 + (q'' - p')\beta - p'' = 0. \tag{10.15}$$

Como los coeficientes en (10.15) son enteros y la fracción continua que define a β (10.13) es infinita, β es un irracional cuadrático.

Del mismo modo, vemos que

$$\alpha = \frac{p_{L-1}\beta + p_{L-2}}{q_{L-1}\beta + q_{L-2}}, \tag{10.16}$$

y despejando β en términos de α,

$$\beta = \frac{p_{L-2} - q_{L-2}\alpha}{q_{L-1}\alpha - p_{L-1}}.$$

Reemplazando en (10.15) y eliminando fracciones, nos queda una ecuación de la forma

$$a\alpha^2 + b\alpha + c = 0,$$

de coeficientes enteros. Más aún, como α es irracional, $a \neq 0$ y $b^2 - 4ac > 0$.

Resumiendo, una fracción continua periódica representa a un irracional cuadrático. La recíproca fue demostrada por Lagrange en 1770, aunque no veremos los detalles de la demostración pues son un poco más complicados:

> *Cada fracción continua periódica representa a un irracional cuadrático y recíprocamente, la fracción continua asociada a un irracional cuadrático es periódica.*

✍ La técnica que acabamos de usar para escribir β en términos de sí mismo en la ecuación (10.14), es similar a la usada para pasar de decimal periódico a fracción que vimos en el ejercicio 7.11, pero vale la pena recordarla en el contexto actual.

Si la expresión decimal de x es

$$x = 0.\underbrace{a_1a_2\ldots a_r}_{\text{período}}a_1a_2\ldots = 0.\bar{a},$$

donde $a = a_1a_2\ldots a_r$ (análoga a (10.13)), tenemos

$$10^r x = a.\bar{a} = a + x,$$

obteniendo una expresión de x en términos de sí mismo (análoga a (10.14)).

El argumento sigue poniendo $\left(10^r - 1\right)x = a$ y luego

$$x = \frac{a}{10^r - 1} = \frac{a_1a_2\ldots a_r}{\underbrace{99\ldots9}_{r \text{ nueves}}},$$

dando lugar a la receta *período dividido tantos* 9 *como cifras tenga el período*.

Si la ecuación (10.11) tiene una solución irracional, el discriminante $D = b^2 - 4ac$ debe ser positivo pero no cuadrado perfecto, y la ecuación tiene dos raíces irracionales distintas, una de la forma

$$\alpha = \frac{P + \sqrt{D}}{Q}$$

y la otra de la forma

$$\alpha' = \frac{P - \sqrt{D}}{Q},$$

Ambas raíces son irracionales cuadráticos, y se dice que son *conjugadas entre sí*.

Es decir, todo irracional cuadrático es de la forma

$$\frac{P + s\sqrt{D}}{Q}, \tag{10.17}$$

donde P, D y Q son enteros, s es 1 o -1, $D > 0$ no es un cuadrado perfecto y podemos suponer $Q > 0$.

E 10.10. Todo irracional de la forma (10.17) satisface la ecuación

$$Q^2x^2 - 2PQx + P^2 - D = 0,$$

y por lo tanto es un irracional cuadrático. ✂

E 10.11. Supongamos que la fracción continua entera infinita asociada al irracional α es de la forma (10.12), y que β está definido por (10.13) y se satisface (10.16).

a) Encontrar coeficientes P_1, Q_1, D_1 y s_1, tales que

$$\beta = \frac{P_1 + s_1\sqrt{D_1}}{Q_1},$$

en términos de los coeficientes $a_L, \ldots, a_{L+k-1}$, es decir, de modo que β quede en la forma (10.17).

b) Encontrar coeficientes P, Q, D y s, tales que

$$\alpha = \frac{P + s\sqrt{D}}{Q},$$

en términos de los coeficientes $a_0, a_1, \ldots, a_{L-1}, a_L, \ldots, a_{L+k-1}$. ✂

E 10.12. Teniendo en cuenta los resultados del ejercicio 10.11 y las notaciones allí usadas:

a) Construir una función `fcaic` (fracción continua a irracional cuadrático) para calcular los coeficientes P, Q, D y s en (10.17), si se sabe que la fracción continua de α es de la forma (10.12) y están dados el anteperíodo $[a_0, a_1, \ldots, a_{L-1}]$ y el período $[a_L, \ldots, a_{L+k-1}]$.

Por ejemplo, `fcaic([1,2],[3,4])` → `(4, 4, 3, 1)`, representando el irracional cuadrático $(4+\sqrt{3})/4$.

b) Una vez obtenidos numéricamente los coeficientes de los apartados *a*) y *b*) del ejercicio 10.11, ¿cómo se los podría simplificar de modo de achicar su valor absoluto lo más posible?

Por ejemplo, $(18+\sqrt{396})/12 \to (3+\sqrt{11})/2$.

✍ No se conocen algoritmos eficientes para descomponer $n \in \mathbb{N}$ como producto de un cuadrado por un número libre de cuadrados, es decir, poner $n = x^2 y$ donde y no tiene factores primos repetidos.

Observar también la relación con la función μ de Moebius (proposición 13.3). ✂

Si bien encontrar la fracción continua asociada a un irracional como hemos hecho en el ejercicio 10.9 es factible desde el punto de vista teórico, desde el punto de vista computacional no se puede realizar en general porque tendríamos que trabajar con precisión «infinita». En el caso de los irracionales cuadráticos se puede soslayar esta dificultad, y usar sólo operaciones enteras para calcular la fracción continua. Los próximos ejercicios apuntan a ese fin.

✍ El libro de Redmond (1996) tiene el desarrollo en detalle de los temas que siguen, así como muchos ejercicios que no necesitan de computadora.

E 10.13. Sea α un irracional cuadrático de la forma (10.17) tal que $Q \mid D - P^2$ y sea $z \in \mathbb{Z}$.

Si definimos

$$P_1 = Qz - P, \quad Q_1 = \frac{D - P_1^2}{Q}, \quad \beta = \frac{P_1 + s\sqrt{D}}{Q_1},$$

entonces Q_1 es entero no nulo y β es un irracional cuadrático tal que

$$\beta\,(\alpha - z) = 1. \tag{10.18}$$

En particular, el inverso multiplicativo de un irracional cuadrático es otro irracional cuadrático. ✁

E 10.14. Con las notaciones anteriores, sea α un irracional cuadrático de la forma (10.17), donde ahora $Q > 0$ y $Q \mid D - P^2$, y pongamos

$$S = \left\lfloor s\sqrt{D} \right\rfloor, \quad P_0 = P, \quad Q_0 = Q, \quad a_0 = \lfloor \alpha \rfloor = \left\lfloor \frac{P_0 + S}{Q_0} \right\rfloor,$$

y para $k \geq 1$,

$$P_k = Q_{k-1}a_{k-1} - P_{k-1}, \quad Q_k = \frac{D - P_k^2}{Q_{k-1}}, \quad a_k = \left\lfloor \frac{P_k + S}{Q_k} \right\rfloor.$$

Entonces $[a_0, a_1, a_2, \ldots]$ es la expresión de α como fracción continua simple.

Sugerencia: Recordar el esquema general (10.10) (¿qué son α_k y ρ_k en este caso?) y los ejercicios 10.13 y 5.3. ✁

E 10.15. Definir una función `icafc(P, Q, s, D, n)` (irracional cuadrático a fracción continua) que construya los $n+1$ primeros coeficientes de la fracción continua del irracional cuadrático α en (10.17).

Por ejemplo,

```
>>> icafc(4, 4, 3, 1, 10)
[1, 2, 3, 4, 3, 4, 3, 4, 3, 4, 3]
```
✁

E 10.16. Modificar la función del ejercicio 10.15 para mostrar en listas separadas anteperíodo y período hasta un máximo combinado de $n+1$ coeficientes.

Por ejemplo:

```
>>> icafc(1, 2, 345, 1, 8)
No se encontró período
[9, 1, 3, 1, 2, 3, 2, 1, 3]
>>> icafc(1, 2, 345, 1, 15)
[[9], [1, 3, 1, 2, 3, 2, 1, 3, 1, 17]]
```
✁

Cabe preguntarse cuándo la fracción continua de un irracional cuadrático es puramente periódica.

Ya hemos visto un ejemplo:

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

En general, decimos que un irracional cuadrático α es *reducido* si $\alpha > 1$ y su conjugado satisface $-1 < \alpha' < 0$, y es posible demostrar:

10.1. Propiedad. *La fracción continua de un irracional cuadrático α es puramente periódica si y sólo si α es reducido.*

E 10.17. Verificar la propiedad 10.1 pa'unlao y pa'lotro con las funciones de los ejercicios 10.12 y 10.16, considerando también casos donde el irracional es negativo o menor que 1. ✂

Otros irracionales cuadráticos particularmente interesantes son los de la forma $\sqrt{D}$, es decir en (10.17) tenemos $P = 0$, $Q = 1$ y $s = 1$. En este caso es posible demostrar:

10.2. Propiedad. *Si $D \in \mathbb{N}$ no es un cuadrado perfecto, entonces*

$$\sqrt{D} = [a_0, \overline{a_1, a_2, \ldots, a_n}],$$

donde

$$a_n = 2a_0, \tag{10.19a}$$

y el período es casi «capicúa» (bah, simétrico):

$$a_i = a_{n-i} \quad \textit{para } i = 1, \ldots n-1. \tag{10.19b}$$

E 10.18. En este ejercicio nos familiarizamos con la propiedad 10.2.

a) Comprobar la propiedad usando la función del ejercicio 10.16 para encontrar la fracción continua asociada a $\sqrt{D}$ para varios valores de $D \in \mathbb{N}$ ($\sqrt{D} \notin \mathbb{N}$).

b) La recíproca no es cierta: encontrar un irracional cuadrático cuya fracción continua satisface las condiciones (10.19) pero no es la raíz cuadrada de un natural. ✂

Es posible ver que la recíproca de la propiedad 10.2 es válida si se amplía el dominio de los números considerados, obteniendo una caracterización de las fracciones continuas de las raíces cuadradas de racionales que son irracionales:

10.3. Propiedad. *El irracional positivo ξ tiene una fracción continua periódica entera de la forma (10.19) si y sólo si $\xi = \sqrt{p/q}$ donde p y q son naturales y $p/q > 1$ (y $\sqrt{p/q}$ es irracional).*

E 10.19. *a*) ¿Por qué en la propiedad 10.3 se pide $p/q > 1$?

b) Ver que si p y q son naturales, podemos encontrar naturales D y Q tales que $\sqrt{p/q} = \sqrt{D}/Q$, y recíprocamente.

c) Verificar la propiedad 10.3 (para un lado y para el otro), usando las funciones de los ejercicios 10.12 y 10.16. ✂

E 10.20. Cuando el irracional cuadrático es simplemente $\sqrt{D}$, se puede demostrar que en el algoritmo del ejercicio 10.14 la parte periódica termina en cuanto $Q_k = 1$ para $k > 0$ (en este caso $Q_0 = 1$). Modificar las función del ejercicio 10.16 reflejando esta propiedad.

Usando la nueva función, encontrar los valores de D que dan los períodos de mayor longitud si $D \le 1000$. ✂

11. El método RSA

Rivest, Shamir y Adleman desarrollaron en 1977 uno de los primeros métodos de *clave pública* para encriptar mensajes, que debido a su éxito es conocido como *método RSA*. En los métodos de clave pública, la forma de *cifrar* o *encriptar* el mensaje es conocida por todos, mientras que la forma de *descifrar* el mensaje es secreta: son *asimétricos*.

Otros métodos son *simétricos*: tanto el que envía como el que recibe el mensaje saben cómo cifrar o descifrar. Estos métodos son en general más rápidos que el RSA, por lo que éste se usa a veces en una primera fase para que quienes se comuniquen se pongan de acuerdo en las claves a usar para comunicar posteriormente el grueso del mensaje.

Acá presentamos una versión sencilla del método, tratando de dar una idea del uso de primos y factorización en criptografía. En la práctica se usan versiones mucho más sofisticadas.

Un buen libro de referencia al alcance de mortales es el de Crandall y Pomerance (2005), que incluye cuestiones del tratamiento numérico de números enteros, incluyendo aspectos del método RSA. Wikipedia tiene bastante información sobre el método, incluyendo referencias.

11.1. RSA: la receta básica

Esencialmente el método RSA consta de los siguientes ingredientes:

1. Se eligen dos primos p y q, «grandes» y distintos.
2. Se define $n = p \times q$ y $m = \phi(n)$, donde ϕ es la función de Euler (recordar el ejercicio 9.19, p. 61),
$$\phi(n) = \#(\{z \in \mathbb{Z} : 1 \le z \le n,\ \mathrm{mcd}(n,z) = 1\}).$$
3. Se consideran dos enteros, e (encriptar) y d (descifrar) tales que
$$e \times d \equiv 1 \pmod{m}. \tag{11.1}$$
Necesariamente e y d deben ser coprimos con m.
4. Llamemos M al «mensaje» a cifrar. Nosotros tomaremos $M \in \mathbb{N}$ (eventualmente pasando letras a números), tal que $1 \le M < n$ con $\mathrm{mcd}(M,n) = 1$ (tal vez dividiendo el mensaje en varias partes).

   ✍ En la práctica, el texto a codificar se transforma en el número M teniendo ciertos cuidados, se produce el número E y cuando se lo descifra (obteniendo M) se lo vuelve a transformar en texto.

   ✍ En el ejercicio 11.5 analizamos el caso $\mathrm{mcd}(M,n) > 1$.
5. El que envía el mensaje M conoce n y e, y envía el mensaje encriptado $E \equiv M^e \pmod{n}$.
6. El que recibe el mensaje E conoce n y d y lo convierte a $M' \equiv E^d \pmod{n}$.

E 11.1. Teniendo en cuenta las definiciones y notaciones anteriores:

a) Demostrar que $M' \equiv M \pmod{n}$.
Ayuda: recordar el teorema de Fermat-Euler (ejercicio 9.20, p. 62).

b) ¿Es la función $M \to E$ inyectiva (uno a uno) sobre los M tales que $\mathrm{mcd}(M,n) = 1$ (y $1 \le M < n$)? ✂

La bondad del método depende de varios factores:

- Los cálculos tienen que hacerse en forma rápida y sencilla. Tal vez el más importante sea el del resto de a^b cuando dividido por c

que aparece en distintas partes y puede hacerse eficientemente usando una variante de la regla de Horner, como hemos visto en el ejercicio 9.12.

✍ Recordar que Python tiene la función `pow(a, b, c)`.

- Debe ser sencillo encontrar d dados n, $\phi(n)$ y e: esto lo hacemos en el ejercicio 11.2.
- Si sólo se conocen n, e y E, debe ser difícil calcular M.

  Por supuesto, esto es sencillo si también se conoce $\phi(n)$ como hicimos en el apartado 6 de la «receta básica» y el ejercicio 11.1. En el ejercicio 11.3 veremos que el cálculo de $\phi(n)$ es equivalente a la factorización de n cuando $n = p \times q$.

  Sin embargo, es posible que haya algún otro método de calcular M (conociendo n, e y E) sin tener conocimiento previo de $\phi(n)$.

  Recordemos que n y e son conocidos por «todos», mientras que $\phi(n)$ permanece secreto y es prudente tirar d, p y q.

  No veremos algoritmos avanzados de factorización pues nos llevaría mucho tiempo.
- En la práctica los primos p y q deben satisfacer ciertas condiciones para que n no se pueda factorizar fácilmente.

  Por ejemplo, es bueno elegir aleatoriamente los valores de p y q. Como no hay una «fórmula» para obtener primos, se elige un número al azar y se verifica si es primo. No es difícil encontrar un primo, como sabemos por el teorema de los números primos (ejercicio 7.3).

  No veremos generación de números aleatorios ni algoritmos avanzados para determinar si un número es primo o no, lo que está fuera del alcance de estas notas (recordar el ejercicio 6.1 y los comentarios al principio de la sección 7).

Además de sistemas específicos para criptografía, muchos sistemas computacionales generales, como los libremente disponibles OpenSSL y Java, o los comerciales como Mathematica y Maple tienen implementaciones elaboradas del método RSA. Acá nos contentaremos con presentar versiones elementales de algunos algoritmos usando el lenguaje Python.

11.2. Mezclando ingredientes

En esta sección suponemos que p y q son dos primos distintos, $n = p \times q$ y $m = \phi(n)$.

E 11.2. Dados n, m y e, con $\mcd(e,m) = 1$, ¿cómo se puede encontrar d de modo de satisfacer la ecuación (11.1)?

Sugerencia: recordar la identidad de Bézout.

Sugerencia si la anterior no alcanza: ver el ejercicio 9.13. ✂

E 11.3. En este ejercicio vemos que el conocimiento de n y $\phi(n)$ es equivalente al conocimiento de p y q, o sea, a la factorización de n.

a) Encontrar $\phi(n)$ en términos de p y q.

b) Determinar p y q en términos de n y $\phi(n)$.

c) Usar el resultado anterior para calcular p y q si

$$n = pq = 493, \quad m = \phi(n) = 448.$$

d) Construir una función para el cálculo de p y q dados n y $m = \phi(n)$ y aplicarlos al caso

$$n = pq = 19\,749\,361\,535\,894\,833,$$
$$m = \phi(n) = 19\,749\,361\,232\,517\,120.$$ ✂

E 11.4. Usar el método RSA cuando $n = 2701$ y $e = 5$ para calcular:

a) E si $M = 1234$,

b) M si $E = 2345$. ✂

E 11.5. En la práctica, no sabemos a priori si $\mcd(M,n) = 1$.

a) Encontrar un ejemplo donde $\mcd(M,n) > 1$.

b) Calcular la probabilidad (casos favorables sobre totales) de que $\mcd(M,n) > 1$ ($n = p \times q$, $0 \le M < n$).

c) Estudiar qué sucede al calcular M^e o E^d (mód n) si $\mcd(M,n) > 1$.

Sugerencia: Calcular los restos módulo p y q usando el «pequeño» teorema de Fermat (ejercicio 9.8, p. 56) y luego usar el teorema chino del resto (ejercicio 9.14, p. 58). ✂

El ejercicio 11.5 muestra una posible aplicación del teorema chino del resto desde la teoría, pero también sugiere una posibilidad práctica: la de descomponer los cálculos para trabajar con números más chicos.

Así, recordando que trabajamos con $n = p \times q$ y $m = \phi(n)$, podemos tratar de descomponer los cálculos para usar sólo operaciones módulo p, q, $p-1$ o $q-1$.

- Cálculo de d como inverso de e módulo $m = (p-1)\times(q-1)$.

  Este cálculo sólo tiene sentido para el receptor del mensaje quien en principio es el único que conoce p y q.

  La dificultad del cómputo yace en que $\mathrm{mcd}(p-1, q-1)$ es un múltiplo de 2, y no se puede usar directamente el teorema del resto chino.

  La factorización de m como producto de factores coprimos no es sencilla, pero podría ser que quitando las potencias de 2 a $p-1$ y $q-1$ quedaran dos números impares y coprimos, digamos $\tilde{p}$ y $\tilde{q}$. En este caso podría usarse una variante del ejercicio 11.6 con tres números: una potencia de 2, $\tilde{p}$ y $\tilde{q}$.

  Otra variante es usar como m a $\mathrm{mcm}(p-1, q-1)$ en vez de $(p-1)\times(q-1)$, disminuyendo un tanto el tamaño de los números que intervienen. El ejercicio 11.7 muestra que esto es válido.

- Cálculo de $E \underset{n}{\equiv} M^e$ y de $M \underset{n}{\equiv} E^d$.

  Aquí los valores de p y q (o $p-1$ y $q-1$) deben conocerse, lo que es inaceptable para quien calcula E. Sólo el cálculo de M a partir de E tendría sentido.

  Si bien quien recibe el mensaje podría eventualmente decodificarlo más eficientemente, quien lo envía no obtiene ganancias (pues desconoce p y q), y es un tanto discutible su conveniencia.

  De cualquier manera, es lo que hacemos en el ejercicio 11.8.

E 11.6. Sean $s_1, s_2, \ldots, s_k$ enteros mayores que 1, coprimos dos a dos, $s = s_1 \times \cdots \times s_k$ y sea $t \in \mathbb{N}$ tal que $\mathrm{mcd}(s,t) = 1$.

a) Encontrar enteros $v_1, v_2, \ldots, v_k$ tales que

$$v_i \times t \equiv 1 \pmod{s_i}, \quad i = 1,2,\ldots,k.$$

b) Encontrar $v \in \mathbb{N}$ tal que

$$v \equiv v_i \pmod{s_i}, \quad i = 1,2,\ldots,k.$$

c) ¿Es cierto que $t \times v \equiv 1 \pmod{s}$?

E 11.7. Sean p y q primos distintos, $n = p \times q$, $m = (p-1) \times (q-1)$, $\lambda = \text{mcm}(p-1, q-1)$.

Ver que si $\text{mcd}(b,n) = 1$, entonces $b^\lambda \underset{n}{\equiv} 1$.

✍ Se trata de un refinamiento del teorema de Fermat-Euler (ejercicio 9.20, p. 62).

✍ La *función de Carmichael* se define como el menor exponente λ tal que $a^\lambda \equiv 1 \pmod{n}$ para todo $a \in \mathbb{N}$ con $\text{mcd}(a,n) = 1$, es decir,

$$\lambda(n) = \text{máx}\,\{o(a) : a \in \mathscr{M}(\backslash)\},$$

donde $o(a)$ es el *orden* dado en la definición 13.6.

Análoga a la función $\phi(n)$, se puede ver que si $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ es la descomposición en primos de n, entonces

$$\lambda(n) = \text{mcm}\left\{\lambda\left(p_1^{\alpha_1}\right), \lambda\left(p_2^{\alpha_2}\right), \ldots, \lambda\left(p_k^{\alpha_k}\right).\right\}$$

E 11.8. Queremos calcular $u^v \pmod{n}$, donde $\text{mcd}(u,n) = 1$ y $n = p \times q$ es el producto de dos primos, haciendo las potencias de la forma $a^b \pmod{c}$ cuando c es p o q pero no n.

a) Definiendo v_1 y v_2 mediante

$$v_1 \underset{p-1}{\equiv} v, \quad v_2 \underset{q-1}{\equiv} v,$$

y u_1, u_2 por

$$u_1 \underset{p}{\equiv} u, \quad u_2 \underset{q}{\equiv} u,$$

ver que

$$u^v \underset{p}{\equiv} u_1^{v_1}, \quad u^v \underset{q}{\equiv} u_2^{v_2}.$$

b) Usando las ecuaciones (9.14), si $1 = ap + bq$, entonces

$$u^v \underset{n}{\equiv} u_1^{v_1} bq + u_2^{v_2} ap.$$

12. Números de Carmichael

Como ya hemos mencionado, Agrawal y otros demostraron la existencia de algoritmos relativamente eficientes (polinomiales en la jerga computacional) para determinar si un número es primo, pero en la práctica se usan otros métodos más rápidos, generalmente probabilísticos.

Recordemos que el «pequeño» teorema de Fermat (ejercicio 9.8, p. 56) establece que si n es primo y $b \in \mathbb{Z}$ entonces

$$b^{n-1} \underset{n}{\equiv} 1 \quad \text{si } n \nmid b, \tag{12.1}$$

o, equivalentemente, si n es primo y $b \in \mathbb{Z}$ es arbitrario,

$$b^n \underset{n}{\equiv} b, \tag{12.2}$$

dando lugar una prueba sencilla para determinar si un número es primo:

> *si para un dado $n \in \mathbb{N}$ encontramos $b \in \mathbb{N}$ tal que $0 < b < n$ y $b^{n-1} \not\equiv 1 \pmod{n}$, n no puede ser primo.*

Claro que esta prueba no es perfecta.

Por un lado, si al aplicar el criterio encontramos b y determinamos que n no es primo, no tenemos forma de conocer dos factores no triviales de n.

Por otro lado, no parece razonable recorrer *todos* los valores de b entre 2 y $n-1$ para encontrar algún b. De cualquier forma, un test rápido es probar con unas cuantas bases b, y si el número pasa todas las pruebas se puede intentar con otros tests más concluyentes para determinar si se trata de un primo.

Miremos con algo más de detalle las ecuaciones (12.1) y (12.2).

E 12.1. Si $n \in \mathbb{N}$, $n \geq 2$, entonces

$$(n-1)^{n-1} \underset{n}{\equiv} \begin{cases} 1 & \text{si } n \text{ es impar,} \\ -1 & \text{si } n \text{ es par.} \end{cases}$$

Por lo tanto:

- Si n es par, $n \geq 4$, y $b = n-1$, la ecuación (12.1) nunca se cumple.
- Cuando n es impar, basta considerar la ecuación (12.1) sólo para $1 < b < n-1$. ✂

E 12.2. Sean n y b enteros, $n > 1$.

a) Si vale (12.1), entonces vale (12.2).

b) Si vale (12.2), entonces vale (12.1) si y sólo si $\mathrm{mcd}(n, b) = 1$.

c) Encontrar n y b, $1 < b < n-1$, tales que vale (12.2) y no vale (12.1).

d) Encontrar n y b, con $1 < b < n-1$ y $\mcd(b,n) = 1$, tales que n no es primo y vale (12.2) (y por lo tanto también (12.1)). ✂

En el ejercicio 12.2 vimos que existen números b y n tales que $1 < b < n-1$, $b^{n-1} \underset{n}{\equiv} 1$ y n no es primo. Algo sorprendentemente, la recíproca del «pequeño» teorema de Fermat no es cierta, esto es, existen números n que no son primos tales que $b^{n-1} \underset{n}{\equiv} 1$ para *cualquier base* b (con, necesariamente, $\mcd(n,b) = 1$). El primero de ellos es $561 = 3 \times 11 \times 17$, mencionado por R. D. Carmichael en sus estudios (1910; 1912) y de allí el nombre de los números. Es un poco injusto que no se nombre también a Korselt (1899), quien había estudiado estos números sin dar ejemplos, tal vez buscando propiedades para demostrar que no existían.

El ejercicio 12.2 también nos trae la duda si definir los números de Carmichael a partir de (12.1) o a partir de (12.2): algunos autores toman la primera y otros la segunda. Acá adoptamos la definición en Crandall y Pomerance (2005), y veremos en el ejercicio 12.6 que las definiciones son equivalentes.

12.1. Definición. Un entero impar positivo n es un *número de Carmichael* si no es primo y vale (12.2) para todo b con $1 < b < n-1$. ♣

E 12.3. Encontrar los números de Carmichael menores a 100 000.

✍ Los números 561, 1105, 1729, 2465, 2821, 6601 y 8911 fueron encontrados por Šimerka (1885), mucho antes de los estudios de Carmichael o el mismo Korselt. ✂

12.2. Teorema (Korselt). *Un número n entero, positivo y compuesto es un número de Carmichael si y sólo si es libre de cuadrados y para cada primo p que divide a n se tiene $p-1 \mid n-1$.*

Veamos la demostración en Crandall y Pomerance (2005) pautada como ejercicios.

E 12.4. Sea n un número de Carmichael (según la definición 12.1), esto es, n es impar, compuesto y $b^n \underset{n}{\equiv} b$ para todo $1 < b < n-1$, e indiquemos con p a un primo que divide a n.

a) $p^2 \nmid n$, es decir, n es libre de cuadrados.

b) $b^n \underset{n}{\equiv} b$ implica $b^n \underset{p}{\equiv} b$.

c) Si $1 < b < p$, entonces $b^n \underset{p}{\equiv} b$ y $b^{n-1} \underset{p}{\equiv} 1$.

d) $(p-1) \mid (n-1)$.

Sugerencia: Usar el teorema de Gauss (teorema 13.15, p. 95). ✂

E 12.5. Supongamos ahora que n es un entero positivo, compuesto, libre de cuadrados, tal que $(p-1) \mid (n-1)$ para todo primo p que divide a n.

Como en el ejercicio anterior, en lo que sigue p indica un número primo que divide a n.

a) Si $\text{mcd}(b,n) = 1$, entonces $b^{n-1} \underset{n}{\equiv} 1$ y $b^n \underset{n}{\equiv} b$.

Ayuda: calcular $b^{p-1} \pmod{p}$.

b) $p^n \underset{n}{\equiv} p$.

Ayuda: Si q es otro primo que divide a n...

c) $b^n \underset{n}{\equiv} b$ para todo b.

Ayuda: recordar lo visto en *a*).

d) ¿Dónde se usa que n es libre de cuadrados? ✂

Repasando el ejercicio 12.2, la definición 12.1 y el teorema 12.2 y su demostración, el siguiente ejercicio nos muestra la equivalencia de las definiciones de números de Carmichael.

E 12.6. Si n es un número entero positivo, impar y compuesto, son equivalentes:

a) $b^n \underset{n}{\equiv} b$ para todo b con $1 < b < n-1$.

b) $b^{n-1} \underset{n}{\equiv} 1$ para todo b coprimo con n, $1 < b < n-1$. ✂

Resumiendo, el «pequeño» teorema de Fermat nos da dos opciones para verificar si un número es primo según se use la ecuación (12.1) o la (12.2). Si p es primo, p satisface ambas alternativas:

$$b^{n-1} \underset{p}{\equiv} 1 \quad \text{para todo } 1 < b < p-1,$$
$$b^n \underset{p}{\equiv} b \quad \text{para todo } 1 < b < p-1.$$

Por el ejercicio 12.2.*b*), un número de Carmichael satisface la segunda condición pero no la primera. Por lo tanto un test para determinar si un número es de Carmichael es ir verificando para cada b, $1 < b < n-1$, si vale (12.1) y en caso contrario comprobar si vale (12.2).

Sin embargo es mucho más rápido (con las herramientas que tenemos) verificar si n es primo buscando factores que no superan $\sqrt{n}$ y luego verificar si vale (12.2).

En fin, la caracterización de Korselt y propiedades como las de los ejercicios 12.7 o 12.8 dan lugar a otras posibilidades para construir o verificar si un número es de Carmichael.

E 12.7. Todo número de Carmichael es impar y tiene al menos tres factores primos.

Ayuda: recordar el ejercicio 12.1 y dividir $pq-1$ por $q-1$ como polinomios tomando $p<q$. ✂

E 12.8. Sea n de Carmichael y p un primo que lo divide.

a) $(p-1)\mid((n/p)-1)$.

Ayuda: hacer la división como polinomios.

b) $p \le n/p$.

c) $p<\sqrt{n}$. ✂

Si bien los números de Carmichael son «ralos», habiendo sólo 2163 menores que 25×10^9, Alford, Granville y Pomerance (1994) demostraron que hay infinitos números de Carmichael. Más aún, Wright (2013) demostró que si a y b son coprimos, hay infinitos números de Carmichael n tales que $n\equiv a \pmod b$, y en 2016 que existen infinitos valores R tales que existen infinitos números de Carmichael con exactamente R factores primos.

E 12.9. Encontrar un número de Carmichael con 4 factores primos. ✂

13. Algo de teoría

Tratando de hacer estas notas bastante autocontenidas, en esta sección vemos algunos resultados teóricos que culminan en el teorema de Gauss (teorema 13.15).

Como es conveniente apelar a nociones más avanzadas como grupos, cuerpos y cocientes por relaciones de equivalencia, separamos estos resultados del grueso de las notas.

Así, en lo que sigue suponemos al lector familiarizado con la definición de $\mathbb{Z}_n$ (para n entero mayor que 1), como el cociente de $\mathbb{Z}$ por la relación de

equivalencia $\underset{n}{\equiv}$ munido de las operaciones de suma y producto (módulo n), y no haremos distinciones entre las clases de equivalencia y sus representantes.

Usaremos también la notación

$$\mathbb{Z}_n^* = \mathbb{Z}_n \setminus \{0\} = \{k \in \mathbb{Z}_n : k \neq 0\}.$$

✍ $\mathbb{Z}_n$ forma un *anillo conmutativo* con las operaciones módulo n.

✍ Textos más recientes usan las notaciones $\mathbb{Z}/n\mathbb{Z}$ o $\mathbb{Z}/(n)$ en vez de $\mathbb{Z}_n$, para evitar confusiones con los enteros n-ádicos.

La siguiente proposición es una reescritura de lo visto en los ejercicios 9.19 y 9.20.

13.1. Proposición. *Sea n es un entero mayor que 1, y $\mathscr{M}(n)$ el subconjunto de $\mathbb{Z}_n$ definido por*

$$\mathscr{M}(n) = \{k \in \mathbb{Z}_n : \operatorname{mcd}(k,n) = 1\}.$$

Entonces

a) $\mathscr{M}(n)$ es un grupo multiplicativo (con la multiplicación en $\mathbb{Z}_n$).

b) $\#\big(\mathscr{M}(n)\big) = \phi(n)$.

c) Si $z \in \mathscr{M}(n)$ entonces $z^{-1} = z^{\phi(n)-1}$.

13.2. Corolario. *$\mathbb{Z}_n$ es un cuerpo $\Leftrightarrow$ n es primo.*

Demostración. El inverso multiplicativo de $a \in \mathbb{Z}_n$ es un elemento b tal que $a \times b \underset{n}{\equiv} 1$, y por lo tanto debe ser $\operatorname{mcd}(a,n) = 1$. Entonces $\mathbb{Z}_n$ es cuerpo si y sólo si $\mathscr{M}(n) = \mathbb{Z}_n \setminus \{0\}$. Para todo $n > 1$ vale $\mathscr{M}(n) \subset \mathbb{Z}_n^*$ y la igualdad se verifica si y sólo si $\#(\mathscr{M}(n)) = \phi(n)$ coincide con $\#(\mathbb{Z}_n) - 1 = n - 1$, es decir, si y sólo si n es primo. ✓

13.3. Proposición (inversión de Moebius). *La función de Moebius μ se define para $n \in \mathbb{N}$ por:*

$$\mu(n) = \begin{cases} 1 & \text{si } n = 1, \\ (-1)^k & \text{si } n \text{ es libre de cuadrados y tiene } k \text{ factores primos}, \\ 0 & \text{en otro caso.} \end{cases}$$

Considerando sólo divisores positivos en las siguientes sumas, tenemos:

a)

$$\sum_{d:d|n} \mu(d) = \begin{cases} 0 & \text{para } n > 1, \\ 1 & \text{si } n = 1. \end{cases}$$

b) Fijemos $n \in \mathbb{N}$ y consideremos f y g definidas para los divisores positivos de n de modo que

$$f(m) = \sum_{d:d|m} g(d) \quad \text{para todo } m \text{ que divide a } n.$$

Entonces

$$g(n) = \sum_{d:d|n} \mu(d) f(n/d).$$

✍ A. F. Moebius introdujo la fórmula en 1832.

Posteriormente la fórmula fue extendida a conjuntos parcialmente ordenados (los naturales están parcialmente ordenados por $a \mid b$), y G. C. Rota dio en 1964 una interpretación más general en un trabajo fundamental de profunda influencia. Así, en combinatoria la fórmula es esencialmente el «principio de inclusión-exclusión».

Demostración. Sea $n > 1$ y $p_1^{\alpha_1} \times p_2^{\alpha_2} \times \cdots \times p_m^{\alpha_m}$ su descomposición en primos.

Si $d \mid n$, y está libre de cuadrados, entonces la descomposición de d en primos tiene sólo algunos de los p_i y con exponente 1 o 0 (lo que vale también para $d = 1$), por lo que hay una relación biyectiva entre estos divisores y los subconjuntos de k elementos de $\{p_1, p_2, \ldots, p_m\}$. Como $\mu(d)$ depende sólo de la cantidad de primos elegidos (si d está libre de cuadrados),

$$\sum_{d:d|n} \mu(d) = \sum_{k=0}^{m} (-1)^k \binom{m}{k} = ((-1)+1)^m = 0.$$

Para la segunda parte,

$$\sum_{d:d|n} \mu(d) f(n/d) = \sum_{d:d|n} \mu(d) \left(\sum_{d':d'|(n/d)} g(d') \right).$$

Observamos que las condiciones $d \mid n$ y $d' \mid (n/d)$ son equivalentes a $(dd') \mid n$ y entonces también equivalentes a $d' \mid n$ y $d \mid (n/d')$. Por lo

tanto,

$$\begin{aligned}\sum_{d:d|n}\mu(d)f(n/d) &= \sum_{d':d'|n} g(d')\left(\sum_{d:d|(n/d')}\mu(d)\right)\\ &= g(n)\left(\sum_{d:d|1}\mu(d)\right)+\sum_{\substack{d':d'|n\\ d'<n}} g(d')\left(\sum_{d:d|(n/d')}\mu(d)\right)\\ &= g(n)+\sum_{\substack{d':d'|n\\ d'<n}} g(d')\left(\sum_{d:d|(n/d')}\mu(d)\right) = g(n). \qquad \checkmark\end{aligned}$$

El ejercicio 9.21 y la proposición 13.3 nos llevan a:

13.4. Corolario.

$$\phi(n)=\sum_{d:d|n}\mu(d)\frac{n}{d}.$$

13.5. Proposición. *Sean:*

- *K un cuerpo,*
- *P un polinomio con coeficientes en K y de grado positivo, lo que denotamos con $P\in K[x]$ y $\operatorname{gr}(P)\geq 1$ respectivamente.*
- *$\alpha\in K$.*

a) Son equivalentes:

i) $P(\alpha)=0$, i. e., α es raíz de P.

ii) Existe $Q\in K[x]$ tal que $P(x)=(x-\alpha)Q(x)$.

b) Si valen las anteriores, $\operatorname{gr}(Q)<\operatorname{gr}(P)$.

c) Un polinomio de grado n no puede tener más de n raíces.

Demostración. Las demostraciones son las conocidas para el caso de los reales o los complejos, usando el algoritmo de la división para demostrar que *a.i)* implica *a.ii)*, poniendo

$$P(x)=(x-\alpha)Q(x)+R(x),$$

donde $\operatorname{gr}(R)=0$, o sea, R es una constante que debe ser nula si vale *a.i)*.

✓

En lo que resta de esta sección llamamos G a un grupo finito conmutativo (abeliano) que pensamos como multiplicativo, con $\#(G) = n$ elementos, e indicamos su elemento neutro por e.

13.6. Definición. Si $a \in G$, el *orden de* a, indicado por $o(a)$, es el menor entero positivo h tal que $a^h = e$. ♣

13.7. Lema. *Para $a \in G$, $o(a)$ está bien definido.*

Demostración. Como G es finito, las sucesivas potencias $a, a^2, a^3, \ldots$ en algún momento se repiten, i. e., existen k y j en $\mathbb{N}$, $j < n$, tales que $a^j = a^k$. Multiplicando miembro a miembro por a^{-j} obtenemos $a^{k-j} = e$, es decir, existe $s \in \mathbb{N}$ tal que $a^s = e$, y por lo tanto $o(a)$ está bien definido. ✓

13.8. Teorema (Lagrange). *Si H es un subgrupo no vacío de G entonces $\#(H) \mid \#(G)$.*

✍ Aunque considerado como francés, Joseph-Louis Lagrange (1736–1813) nació en Turín (Italia) como Giuseppe Lodovico Lagrangia.

Como hemos visto, hizo importantes contribuciones a la teoría de números, pero su obra es mucho más amplia. Fue uno de los fundadores del «cálculo de variaciones» (área relacionada con la mecánica) y las probabilidades, e hizo numerosas contribuciones en otras áreas como astronomía y ecuaciones diferenciales.

Estudiantes de cálculo reconocen su nombre por los «multiplicadores» para encontrar extremos de funciones de varias variables.

Demostración. Para $b \in G$ definamos

$$H_b = \{bh : h \in H\}.$$

Observemos que

$$\#(H_b) = \#(H) \quad \text{para todo } b \in G,$$

pues $h \to bh$ es biyectiva (dado que G es grupo).

Además, si $H_b \cap H_c \neq \emptyset$, existen $x, y \in H$ tales que $bx = cy$, y por lo tanto $b = cyx^{-1} \in H_c$, pues $yx^{-1} \in H$. Luego $H_b = H_c$.

En definitiva, para $b \neq c$ debe ser o bien $H_b = H_c$ o bien $H_b \cap H_c$.

Como $\bigcup_{b \in G} H_b = G$, resulta $\#(H) \mid \#(G)$. ✓

13.9. Corolario. *Con las notaciones anteriores, sean $n = \#(G)$ y $a \in G$.*

a) $o(a) \mid n$, *y en particular,* $o(a) \le n$.

b) Si $m \in \mathbb{Z}$, $m \ge 0$, *entonces* $a^m = e \Leftrightarrow o(a) \mid m$.

c) Si $o(a) = h$ *y* $d = \text{mcd}(h,m)$, *entonces* $o(a^m) = h/d$.

Demostración. Pongamos $h = o(a)$ y $A = \{a, a^2, \ldots, a^{h-1}, a^h = e\}$.

Entonces A es un subgrupo de G y $h = \#(A) \mid n$, lo que demuestra *a)*.

Para *b)*, pongamos $m = qh + r$, con $q \ge 0$ y $0 \le r < h$. Entonces $a^m = e \Leftrightarrow a^r = e$, y como h es un mínimo positivo, $a^m = e \Leftrightarrow r = 0$, es decir $a^m = e \Leftrightarrow h \mid m$.

Finalmente, con $d = \text{mcd}(h,m)$ pongamos $h' = h/d$ y $m' = m/d$. Entonces

$$\left(a^m\right)^{h'} = a^{mh'} = a^{m'h} = \left(a^h\right)^{m'} = e.$$

Si ahora s es tal que $a^{ms} = e$, por *b)* sabemos que $ms = kh$ para algún k, entonces $m's = kh'$ y como $\text{mcd}(m', h') = 1$, $h' \mid s$. ✓

13.10. Definición. G es *cíclico* si existe $a \in G$ tal que $\{a, a^2, \ldots, a^{o(a)}\} = G$. En ese caso decimos que a es un *generador* de G. ♣

13.11. Lema (Pero Grullo). *a es generador de* $G \Leftrightarrow o(a) = \#(G)$.

13.12. Teorema. *Sea* G *finito,* $n = \#(G) \ge 2$. *Son equivalentes:*

a) G *es cíclico.*

b) Para cada $d \in \mathbb{N}$ *con* $d \mid n$, $\#\left(\{x \in G : x^d = e\}\right) = d$.

c) Si $d \in \mathbb{N}$ *y* $d \mid n$, $\#(\{x \in G : o(x) = d\}) = \phi(d)$.

Demostración. Demostramos *a)* ⇒ *b)* ⇒ *c)* ⇒ *a)*.

a) ⇒ *b)*. Sea a un generador de G, $o(a) = n$, y sea k tal que $n = kd$.

$A = \{e, a^k, a^{2k}, \ldots, a^{(d-1)k}\}$ tiene d elementos distintos, pues $a^{jk} = a^{ik}$ con $i < j$ implica $a^{(j-i)k} = e$, y entonces $n \mid (j-i)k$, $d \mid (j-i)$, y como $i < j < d$, necesariamente $j - i = 0$, lo que es una contradicción.

Observemos que cada elemento de A elevado a la d da e.

Para ver que no hay otros, supongamos $b^d = e$ y sea j tal que $b = a^j$, $0 \le j < n$ (a es generador). Entonces $e = b^d = a^{jd}$, y tenemos sucesivamente:

$$dk = n = o(a) \mid jd,$$

$$k \mid j,$$

$$j = uk = (vd + r)k \quad \text{con } v \ge 0 \text{ y } 0 \le r < d,$$

$$b = a^j = \left(a^{dk}\right)^v a^{rk} = \left(a^n\right)^v a^{rk} = a^{rk} \in A.$$

b) ⇒ *c)*. Para cada $d \in \mathbb{N}$ divisor de n, consideremos

$$G_d = \left\{x \in G : x^d = e\right\}.$$

G_d es un subgrupo de G, y por *b)*, $\#(G_d) = d$.
Si $u \in \mathbb{N}$ y $u \mid d$,

$$\{a \in G : o(a) = u\} \subset G_d,$$

y poniendo

$$\beta(d) = \#(\{a \in G : o(a) = d\}),$$

usando el lema 13.9, tendremos

$$d = \#(G_d) = \sum_{u:u|d} \beta(u) \quad \text{para todo } d \in \mathbb{N} \text{ con } d \mid n.$$

Usando la proposición 13.3,

$$\beta(n) = \sum_{d:d|n} \mu(d)\frac{n}{d},$$

y por el corolario 13.4, $\beta(n) = \phi(n)$.

c) ⇒ *a)*. Tomando $d = n$, vemos que $\#(\{x \in G : o(x) = n\}) = \phi(n)$. En particular, existe un elemento de orden n, necesariamente generador. ✓

13.13. Corolario. *Si G es cíclico, entonces tiene exactamente $\phi(n)$ generadores.*

13.14. Teorema. *El grupo multiplicativo de un cuerpo finito es cíclico.*

Demostración. Sean K un cuerpo y $G = K \setminus \{0\}$ su grupo multiplicativo (usualmente denotado por K^*), y pongamos $e = 1$, $n = \#(G)$.

Observemos que $a^n = 1$ para todo $a \in G$ pues $o(a) \mid n$.

Si $d \mid n$ y $n = kd$ tendremos

$$x^n - 1 = \left(x^d - 1\right)\left(x^{(k-1)d} + x^{(k-2)d} + \cdots + 1\right) = \left(x^d - 1\right) Q(x).$$

Como K es un cuerpo, por la proposición 13.5 $x^d - 1$ tiene a lo sumo d raíces, Q tiene a lo sumo $(k-1)d$ raíces y todas las n raíces de $x^n - 1$

son raíces de $x^d - 1$ o de Q. Como $d + (k-1)d = kd = n$, vemos que la ecuación $x^d = 1$ tiene exactamente d raíces.

Por lo tanto se satisface la condición *b)* de la proposición 13.12 y G es cíclico. ✓

Usando el corolario 13.2 tenemos:

13.15. Teorema (Gauss). *Si p es primo, $\mathbb{Z}_p^* = \mathscr{M}(p)$ es cíclico y tiene exactamente $\phi(p-1)$ generadores (llamados* raíces primitivas*).*

- ✍ Se puede ver que para $n > 1$ el grupo multiplicativo $\mathscr{M}(n)$ es cíclico si y sólo si n es 2, 4, p^m o $2p^m$ donde p es un primo impar y $m \in \mathbb{N}$.

  Por otra parte, se puede demostrar que si K es un cuerpo finito entonces $\#(K) = p^m$ donde p es un primo y $m \in \mathbb{N}$, y recíprocamente: dados p y m se puede construir un cuerpo finito K con $\#(K) = p^m$. Estos cuerpos son isomorfos entre sí (para p y m fijos), y se los denota por $\mathrm{GF}(p^m)$ (o $\mathbb{F}_{p^m}$ o $\mathbf{F}_{p^m}$).

  Recordando la proposición 13.1, $\mathscr{M}(p^m)$ no es (isomorfo a) el grupo multiplicativo de $\mathrm{GF}(p^m)$ si $m > 1$ (simplemente porque $\phi(p^m) \neq p^m - 1$). En otras palabras, $\mathbb{Z}_{p^m}$ y $\mathrm{GF}(p^m)$ son bichos muy distintos cuando $m > 1$.
- ✍ El resultado es uno de los muchos que aparecen en el libro *Disquisitiones Arithmeticae* que Gauss escribió en 1798 cuando tenía 21 años. El libro fue publicado en 1801 y contiene dos demostraciones de este resultado.
- ✍ No se conoce un algoritmo sencillo para encontrar un *generador* del grupo $\mathbb{Z}_p^*$, si bien se cuenta con algoritmos más o menos rápidos.

Cuando G es el grupo multiplicativo de un cuerpo K, como es el caso de $G = \mathbb{Z}_p^*$, podemos combinar las demostraciones de los teoremas 13.12 y 13.14, evitando la inversión de Moebius.

Con las notaciones del teorema 13.12, consideremos d tal que $\beta(d) > 0$, sea $a \in G$ tal que $o(a) = d$, y sea

$$A = \left\{a, a^2, a^3, \ldots, a^d\right\}.$$

Los elementos de A son d soluciones distintas de la ecuación $x^d = e$, y sabemos que no hay otras por la proposición 13.5 (a lo sumo hay d soluciones a la ecuación $x^d = e$), lo que es *b)* en el teorema 13.12.

Por el corolario 13.9, $o(a^m) = d/k$ donde $k = \text{mcd}(m,d)$, por lo que $o(a^m) = d \Leftrightarrow \text{mcd}(m,d) = 1$. Por lo tanto, si

$$D = \{b \in G : o(b) = d\},$$

por un lado tendremos $D \subset A$ pues $b \in D \Rightarrow b^d = e$, y por otro

$$\beta(d) = \#(D) = \#(\{m \in \mathbb{N} : 1 \leq m \leq d, \text{mcd}(m,d) = 1\}) = \phi(d).$$

De modo que o bien $\beta(d) = 0$ o bien $\beta(d) = \phi(d)$.
Entonces,

$$n = \#(G) = \sum_{d:d|n} \beta(d) \leq \sum_{d:d|n} \phi(d) = n,$$

donde en la última igualdad usamos la proposición 9.21.

Por lo tanto $\beta(d) = \phi(d)$ para todo divisor d de n, lo que demuestra 13.12.*c*), a partir del cual es sencillo verificar 13.12.*a*).

Referencias

M. Agrawal, N. Kayal y N. Saxena, 2004. PRIMES is in P. *Ann. of Math.*, 160:781–793. (*págs. 32 y 84*)

W. R. Alford, A. Granville y C. Pomerance, 1994. There are infinitely many Carmichael numbers. *Ann. of Math.*, 140:703–722. (*pág. 88*)

R. D. Carmichael, 1910. Note on a new number theory function. *Bulletin of the American Mathematical Society*, 16(5):232–238. doi: 10.1090/s0002-9904-1910-01892-9. (*pág. 86*)

R. D. Carmichael, 1912. On composite numbers P which satisfy the Fermat congruence $a^{P-1} \equiv 1$ mód p. *American Mathematical Monthly*, 19(2):22–27. doi: 10.2307/2972687. (*pág. 86*)

T. Cormen, C. Leiserson, R Rivest y C. Stein, 2009. *Introduction to Algorithms*. MIT Press, 3.ª ed. (*pág. 3*)

R. Crandall y C. Pomerance, 2005. *Prime Numbers: A Computational Perspective*. Springer (New York), 2.ª ed. (*págs. 3, 79 y 86*)

A. Engel, 1993. *Exploring Mathematics with your computer*. MAA. (*págs. 3 y 48*)

E. Gentile, 1991. *Aritmética elemental en la formación matemática*. Red Olímpica. (*págs. 1, 2, 3, 4, 6, 7, 11, 14, 53, 54, 55, 56, 57, 60 y 64*)

G. H. Hardy y E. M. Wright, 2008. *An Introduction to the Theory of Numbers*. Oxford University Press, 6.ª ed. (*págs. 2, 24, 65 y 71*)

D. E. Knuth, 1998. *The art of computer programming. Vol. 2. Fundamental Algorithms*. Addison-Wesley, 3.ª ed. (*págs. 3, 52 y 53*)

A. R. Korselt, 1899. Problème chinois. *L'Intermédiaire des Mathématiciens*, 6:142–143. (*pág. 86*)

I. Niven, H. Zuckerman y H. Montgomery, 1991. *An Introduction to the Theory of Numbers*. J. Wiley & Sons, 5.ª ed. (*pág. 2*)

C. D. Olds, 1963. *Continued Fractions*. MAA. (*págs. 3, 54 y 65*)

D. Redmond, 1996. *Number Theory*. Marcel Dekker. (*págs. 2 y 76*)

R. Rivest, A. Shamir y L. Adleman, 1978. A Method for Obtaining Digital Signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126. doi: 10.1145/359340.359342. (*pág. 79*)

K. H. Rosen, 1993. *Elementary Number Theory and its Applications*. Addison Wesley, 3.ª ed. (*págs. 2 y 49*)

V. Šimerka, 1885. Zbytky z arithmetické posloupnosti (on reminders from arithmetical sequence). *Časopis pro pěstování mathematiky a fysiky*, 14 (5):221–225. (*pág. 86*)

T. Wright, 2013. Infinitely many Carmichael numbers in arithmetic progressions. *Bull. London Math. Soc.*, 45:943–952. (*pág. 88*)

T. Wright, 2016. Factors of Carmichael numbers and a weak k-tuples conjecture. *Journal of the Australian Mathematical Society*, 100(3):421–429. doi: 10.1017/S1446788715000427. (*pág. 88*)