

Recuperatorio del primer trabajo práctico

La conjetura de Goldbach

(*O cómo ganar un millón de dólares*)

Taller de Álgebra 1

Primer cuatrimestre de 2021

En 1742, Christian Goldbach —un matemático alemán— en una carta a Leonard Euler o, más precisamente, en el margen de esa carta¹, hizo una conjetura, que es equivalente a la siguiente:

Conjetura. *Todo número entero par mayor que 2 puede ser escrito como la suma de dos números primos.*

Al día de hoy, nadie sabe si esto es cierto o no, y decidirlo es uno de los problemas más famosos de la teoría de números. Desde los tiempos de Goldbach varios matemáticos se han dedicado a juntar evidencia de que la conjetura es cierta. Han habido, ciertamente, muchos avances: por ejemplo, en 1930 Lev Schnirelmanm probó que todo número par mayor que 2 puede escribirse como la suma de no más que 800 000 números primos, lo que está bastante lejos de la conjetura de Goldbach, pero sin embargo fue una contribución extraordinaria. Mucho más recientemente, en 1995, Olivier Ramaré probó que todo entero par mayor que 4 es suma de 6 o menos números primos. Usando computadoras, se ha verificado que todo entero par mayor que 2 y menor que 4×10^{18} es suma de dos números primos. En el año 2000, con el fin de generar publicidad para el libro *El tío Petros y la conjetura de Goldbach* de Apostolos Doxiadis, el editor británico Tony Faber ofreció un premio de un millón de dólares a aquel que demostrase la conjetura. Nadie reclamó el premio aún...

En este trabajo práctico nos proponemos escribir funciones en Haskell para testear la conjetura y quién sabe, quizás de allí surja una demostración que nos permita hacernos millonarios.

¹En esa época usaban mucho los márgenes.

Ejercicio 1

Escribir la función:

```
satisfaceGoldbach :: Integer -> Bool
```

que recibe un número natural `n` y devuelve `True` si y solo si el `n` es par, mayor que 2 y suma de dos números primos o `False` en caso contrario.

Ejercicio 2

Escribir la función

```
verificarConjeturaHasta :: Integer -> Bool
```

que recibe un número natural `n` par mayor que 2 y devuelve `True` si y solo si la conjetura es cierta para todos los naturales pares mayores que 2 y menores o iguales que `n` o `False` en caso contrario.

La función `satisfaceGoldbach` solo nos dirá si un número satisface o no la condición de la conjetura, pero es posible que queramos conocer *cómo* escribirlo como suma de dos números primos.

Ejercicio 3

Escribir la función

```
descomposicionEnPrimos :: Integer -> (Integer,Integer)
```

que recibe un número natural `n` par mayor que 2 y devuelve un par ordenado `(a,b)` de números primos tales que `a + b == n`.

En general, un número par puede escribirse como suma de dos números primos de muchas formas. Por ejemplo, $20 = 3 + 17 = 7 + 13 = 13 + 7 = 17 + 3$ y $10 = 3 + 7 = 5 + 5 = 7 + 3$. Observemos que estamos haciendo una distinción entre $3 + 7$ y $7 + 3$: por eso diremos que 20 tiene 4 descomposiciones mientras que 10 tiene 3.

Ejercicio 4

Escribir la función

```
numeroDeDescomposiciones :: Integer -> Integer
```

que recibe un número natural n par mayor que 2 y devuelve la cantidad de pares ordenados (a, b) de números primos tales que $a + b == n$.

```
*Main> numeroDeDescomposiciones 88
8
*Main> numeroDeDescomposiciones 10
3
*Main> numeroDeDescomposiciones 4
1
*Main> numeroDeDescomposiciones 123456
2904
```

Ejercicio 5: adicional recuperatorio

Escriba la función

```
másDescomponible :: Integer -> Integer
```

que recibe un natural n par mayor que 2 y devuelve un entero entre 1 y n que tiene más descomposiciones como suma de dos números primos. Si hay «empates», que devuelva el más chico. Por ejemplo:

```
*Main> másDescomponible 40
36
```

porque entre los números del 1 al 40 que son pares y mayores que 2 el que de más formas se puede escribir como suma de dos números primos es el 36: hay 8 formas de hacerlo, a saber:

$$36 = 5 + 31 = 7 + 29 = 13 + 23 = 17 + 19 = 19 + 17 = 23 + 13 = 29 + 7 = 31 + 5.$$

Otro ejemplo:

```
*Main> másDescomponible 20
16
```

porque entre los números del 1 al 20 que son pares y mayores que 2, los números con la mayor cantidad de descomposiciones como suma de dos primos son el 16, el 18 y el 20, que tienen 4 descomposiciones cada una —así que hay un empate— y el 16 es el más chico de ellos. Más ejemplos:

```
*Main> másDescomponible 40
36
*Main> másDescomponible 20
16
*Main> másDescomponible 100
90
*Main> másDescomponible 1000
990
*Main> másDescomponible 10000
9240
```

Condiciones de entrega

El trabajo práctico se debe realizar de manera individual. La entrega consiste en un único archivo `.hs` con las funciones de los ejercicios implementadas, junto con todas las funciones auxiliares que sean necesarias para ejecutarlas. Las funciones deben respetar

la signatura (nombres y parámetros) especificados en cada ejercicio, dado que serán testeadas automáticamente. El archivo que entregan tiene que poder compilarse sólo, y sin llamar a otro módulo.

El nombre del archivo entregado debe tener la forma “turno-nombreapellido.hs”, donde turno debe ser “TM”, “TT” o “TN” dependiendo del turno en el que estén cursando:

- TM: Miércoles 9 hs.
- TT: Viernes 14 hs.
- TN: Miércoles 17:30 hs.

Por ejemplo, un estudiante llamado Juan Pérez que esté cursando en el turno de los viernes debe entregar un archivo llamado “TT-JuanPerez.hs”. La entrega se debe llevar a cabo a través del campus virtual, subiendo el archivo con el código con el mecanismo disponible dentro del espacio del taller en el campus virtual.

Para la resolución del trabajo práctico se deben utilizar exclusivamente los conceptos vistos hasta la clase 5 del cronograma de la materia, inclusive. Se evaluará la corrección de las funciones implementadas, la declaratividad y claridad del código, y que las funciones auxiliares (si las hay) tengan nombres apropiados.

Si tienen dudas o consultas respecto del trabajo práctico, pueden enviar un mail a la lista de docentes algebra1-doc (arroba) dc.uba.ar. No hacer consultas a través del campus virtual ni mandando mails a la lista de alumnos.

Además del código, cada alumno deberá rendir un **coloquio individual**, en el que conversará sobre el trabajo que realizó con el docente que le corrigió y tendrá que responder una serie de preguntas al respecto.

Fecha de entrega: hasta el viernes 18 de junio a las 23:59 hs.