Recuperatorio del primer trabajo práctico
Números pseudoprimos y de Carmichael

Taller de Álgebra 1 - Segundo cuatrimestre de 2022

El *Pequeño Teorema de Fermat* es un teorema que verán más adelante en la teórica de Álgebra I. Este resultado afirma lo siguiente:

Teorema 1. *Sea p un número primo positivo. Entonces, para todo número natural a coprimo con p, p divide a $a^{p-1} - 1$.*

Como consecuencia de esto, tenemos que para todo primo positivo $p \geq 3$, como 2 es coprimo con p, p divide a $2^{p-1} - 1$. Sin embargo, esta propiedad no es exclusiva de los números primos. Hay algunos números naturales compuestos que también la satisfacen, y ellos se llaman *2-pseudoprimos*. En otras palabras, los 2-pseudoprimos son los números naturales compuestos n para los cuales n divide a $2^{n-1} - 1$. Los primeros 2-pseudoprimos son: 341, 561, 645, 1105, 1387, 1729, 1905, ...

Más en general, dado un número natural a, los *a-pseudoprimos* son los números naturales compuestos n para los cuales n divide a $a^{n-1} - 1$. Los primeros 3-pseudoprimos son 91, 121, 286, 671, 703, 949, 1105, 1541, 1729, ... Los primeros 4-pseudoprimos son 15, 85, 91, 341, 435, 451, 561, 645, 703, ...

Finalmente, los *números de Carmichael* son los números naturales compuestos n que son a-pseudoprimos para todo número natural a entre 1 y $n-1$ que sea coprimo con n. Es sabido que existen infinitos números de Carmichael, pero aparecen muy espaciadamente dentro de la sucesión de los números naturales. Los únicos números de Carmichael menores a 100.000 son 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973 y 75361. Los números de Carmichael son muy importantes en Teoría de Números, porque si bien no son números primos, pasan algunos test probabilísticos de primalidad que posiblemente verán más adelante en la teórica de Álgebra I.

Un número n es *libre de cuadrados* cuando para todo divisor $m > 1$ de n se tiene que m^2 no divide a n. En 1899 Korselt demuestra que un número es Carmichael si y solo si n es impar, compuesto, libre de cuadrados y $p - 1$ divide a $n - 1$ para cada primo p que divide a n.

Ejercicios

Se pide resolver en Haskell los siguientes ejercicios:

Ejercicio 1

Escribir la función:

```
sonCoprimos :: Integer -> Integer -> Bool
```

que dados dos números naturales decide si son coprimos. Por ejemplo:

```
*Main> sonCoprimos 12 9
False
*Main> sonCoprimos 1007 1474
True
*Main> sonCoprimos 1 9
True
```

Ejercicio 2

Escribir la función:

```
es2Pseudoprimo :: Integer -> Bool
```

que dado un número natural decide si es 2-pseudoprimo. Por ejemplo:

```
*Main> es2Pseudoprimo 561
True
*Main> es2Pseudoprimo 1387
True
*Main> es2Pseudoprimo 1728
False
```

Ejercicio 3

Escribir la función:

```
cantidad3Pseudoprimos :: Integer -> Integer
```

que dado un número natural m calcula la cantidad de 3-pseudoprimos que hay entre 1 y m inclusive. Por ejemplo:

```
cantidad3Pseudoprimos 100
1
cantidad3Pseudoprimos 671
4
cantidad3Pseudoprimos 702
4
```

Ejercicio 4

Escribir la función:

```
kesimo2y3Pseudoprimo :: Integer -> Integer
```

que dado un número natural k calcula el k-ésimo número que es simuláneamente 2-pseudoprimo y 3-pseudoprimo. Por ejemplo:

```
kesimo2y3Pseudoprimo 1
1105
kesimo2y3Pseudoprimo 4
2701
kesimo2y3Pseudoprimo 6
6601
```

Ejercicio 5

Escribir la función:

```
esCarmichael :: Integer -> Bool
```

que dado un número natural decide si es un número de Carmichael siguiendo la definición. Por ejemplo:

```
esCarmichael 2465
True
esCarmichael 2821
True
esCarmichael 1541
False
```

Ejercicio 6

Escribir la función:

```
esLibreDeCuadrados :: Integer -> Bool
```

que dado un número natural decide si es libre de cuadrados.

```
esLibreDeCuadrados 1
True
esLibreDeCuadrados 45
False
esLibreDeCuadrados 46
True
```

Ejercicio 7

Escribir la función:

```
esCarmichaelK :: Integer -> Bool
```

que dado un número natural decide si es un número de Carmichael siguiendo el criterio de Korselt.

```
esCarmichaelK 2465
True
esCarmichaelK 2821
True
esCarmichaelK 1541
False
```

Condiciones de entrega

Para el RTP1 se respetan los grupos del TP1. Leer la solapa *Evaluación* en el campus virtual del Taller.

Está prohibido subir el código que implementen a repositorios públicos, así como también usar total o parcialmente soluciones implementadas por otros grupos. No está permitida la interacción con otros grupos para discutir las soluciones de los ejercicios propuestos.

No evaluaremos la eficiencia de los algoritmos que propongan para resolver los ejercicios (y por ende, el tiempo en que tardan en ejecutarse las funciones) pero sí la correctitud del código. Algunas funciones que implementen pueden demorar minutos en arrojar el resultado, no se preocupen por el tiempo sino porque devuelvan el valor correcto.

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 solo, y sin llamar a otro módulo. Les pedimos que utilicen como base para escribir el código el archivo Tp1.hs que se encuentra en el campus virtual del Taller.

El archivo entregado debe tener la forma "`apellido1-apellido2-apellido3.hs`", respetando el orden alfabético de los apellidos. Por ejemplo, el grupo formado por los estudiantes María López, Ariel Gómez, y Juan Pérez debe entregar un archivo llamado "`Gomez-Lopez-Perez.hs`".

Además, en las primeras líneas del archivo deben ir los nombres completos comentados, empezando por el apellido y siguiendo por el o los nombres sin comas y la dirección de email. En el ejemplo:

```
-- López María lopezmaria@xxx.com
-- Gómez Ariel gomezariel@yyy.com
-- Pérez Juan perezjuan@zzz.com
```

La entrega se debe llevar a cabo a través del campus virtual, subiendo el archivo con el código con el mecanismo disponible dentro de la solapa *Evaluación* en el campus virtual del taller. Un solo integrante será el encargado de subir el Trabajo Práctico al campus en representación de todo el grupo.

Se deberán utilizar exclusivamente los conceptos vistos hasta ahora (semana 6 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 de la lista de mails de alumnos.

Además del código, deberán rendir un coloquio, en el que se conversará **individualmente con cada integrante del grupo** sobre el trabajo realizado, debiendo responder diversas preguntas sobre lo que entregaron. El día y horario del coloquio lo acordarán por email con el docente que haya corregido su trabajo. Ver sección *Evaluación* en el campus virtual.

Fecha de entrega: Hasta el domingo 20 de noviembre a las 23:55 hs por el campus virtual del Taller en la solapa *Evaluación*.