

Trabajo Práctico

Sopa de números

Taller de Álgebra

Verano 2020

Fecha límite de entrega:

Domingo 8 de marzo hasta las **23:59** hs.

Coloquio: miércoles 11 de marzo

Introducción

Una **sopa de números** es un juego que consiste en descubrir propiedades de un tablero de $n \times m$ ($n, m \geq 1$) en los que en cada posición hay un número entero positivo. Cada posición se identifica con un par (i, j) , en el cual la primera componente corresponde a una fila y la segunda a una columna. A modo de ejemplo, la siguiente figura muestra un tablero de 5×4 en el que el número 13 aparece en la posición $(1, 1)$ y el número 5 aparece en la posición $(4, 3)$.¹

13	12	6	4
1	1	32	25
9	2	14	7
7	3	5	16
27	2	8	18

Un camino en un tablero está dado por una secuencia de posiciones adyacentes en la que solo es posible desplazarse desde una posición dada hacia la posición de su derecha o hacia la que se encuentra debajo. Un camino de longitud l de una sucesión $(a_n)_{n \in \mathbb{N}}$ es un camino en un tablero tal que para todo $1 \leq i \leq l$, el valor i -ésimo del camino es a_i . Siguiendo con el ejemplo, a continuación puede observarse un camino de longitud 5 de la sucesión Fibonacci que empieza en la posición $(2, 1)$ y termina en $(4, 3)$ del tablero.

13	12	6	4
1	1	32	25
9	2	14	7
7	3	5	16
27	2	8	18

Para manipular las sopas de números en Haskell vamos a representar el tablero como una lista de filas de igual longitud que no se repiten. A su vez, cada fila vamos a representarla como una lista de enteros positivos. Las posiciones vamos a representarlas con tuplas de dos números enteros positivos y un camino va a estar dado por una lista de posiciones.

¹Notar que tanto la numeración de las filas como la de las columnas comienzan en 1.

El archivo `alg1-tp.hs` tiene las siguientes definiciones y funciones:

- **type** Conjunto a = [a]
- **type** Fila = [Integer]
- **type** Tablero = [Fila] (en un tablero, todas las filas son distintas y de igual longitud)
- **type** Posicion = (Integer, Integer)
- **type** Camino = [Posicion]
- `cantidadFilas :: Tablero -> Integer`
Devuelve la cantidad filas de un tablero.
- `cantidadColumnas :: Tablero -> Integer`
Devuelve la cantidad columnas de un tablero.
- `valor :: Tablero -> Posicion -> Integer`
Devuelve el valor de una posición de un tablero.
- `posValida :: Tablero -> Posicion -> Bool`
Determina si una posición está dentro de los límites de un tablero.

Ejercicios

Se pide implementar las siguientes funciones en Haskell:

1. `maximo :: Tablero -> Integer`
Devuelve el número más grande de un tablero dado.
2. `masRepetido :: Tablero -> Integer`
Devuelve el número que más veces aparece en un tablero dado. Si hay empate devuelve cualquiera de ellos.
3. `valoresDeCamino :: Tablero -> Camino -> [Integer]`
Devuelve los valores de las posiciones de un camino.
4. `caminoDeCollatz :: Tablero -> Camino -> Integer -> Bool`
Determina si los valores de las posiciones de un camino forman un camino de la sucesión de Collatz iniciada con el número pasado como parámetro. Lothar Collatz planteó una conjetura en 1937 (¡y aún no se demostró!) que dada siguiente sucesión:

$$a_{n+1} = \begin{cases} \frac{a_n}{2} & \text{si } a_n \text{ es par} \\ 3a_n + 1 & \text{si } a_n \text{ es impar} \end{cases}$$

empezando a_1 con cualquier número positivo siempre se llega a 1. Por ejemplo, si $a_1 = 13$, obtenemos la siguiente secuencia de números: $13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

5. `mayorSecuenciaDeCollatz :: Tablero -> Integer -> [Integer]`
Devuelve la lista de los valores de las posiciones del camino más largo del tablero de la sucesión Collatz inicializada con el número positivo pasado como parámetro. Si hay empate devuelve cualquiera de ellas (serán iguales). Si el número pasado como parámetro no existe en el tablero, se devuelve la lista vacía. Si el número está en el tablero, pero no se puede continuar una sucesión Collatz, devuelve la lista formada sólo con el valor del número. (**¡Atención!** Las secuencias de Collatz terminan en el 1. La secuencia $[8, 4, 2, 1, 4]$ no es válida.)
6. `mayorSecuenciaDeCollatzPermutando :: Tablero -> Integer -> [Integer]`
Idem ejercicio anterior, devuelve la lista de los valores de las posiciones del camino de Collatz más largo inicializado con el número positivo pasado como parámetro, pero en este caso de todas las permutaciones posibles de filas del tablero. **Ayuda:** se sugiere resolver primero el problema de construir el conjunto de tableros que resulta de permutar todas las filas del tablero pasado como parámetro y luego llamar a la función del ejercicio anterior con cada tablero del conjunto.

Casos de test

En la siguiente tabla, la evaluación de las expresiones de la columna izquierda debe devolver los valores de la columna derecha. Los argumentos utilizados en estos casos de test (e.g. `sopa1`, `sopa2`, `camino1`) están definidos en `alg1-tp.hs`.

<u>Expresión</u>	<u>Resultado esperado</u>
<code>maximo sopa1</code>	32
<code>maximo sopa2</code>	64
<code>masRepetido sopa1</code>	2
<code>masRepetido sopa2</code>	8 o 4
<code>valoresDeCamino sopa1 camino1</code>	[16,8,4]
<code>valoresDeCamino sopa1 camino2</code>	[12,1,32,16,5]
<code>valoresDeCamino sopa2 camino1</code>	[4,6,1]
<code>valoresDeCamino sopa2 camino2</code>	[5,16,8,4,2]
<code>caminoDeCollatz sopa1 camino1 16</code>	True
<code>caminoDeCollatz sopa3 camino1 11</code>	False
<code>mayorSecuenciaDeCollatz sopa1 32</code>	[32,16,8,4,2]
<code>mayorSecuenciaDeCollatz sopa2 8</code>	[8,4,2,1]
<code>mayorSecuenciaDeCollatz sopa3 7</code>	[7,22,11,34,17]
<code>mayorSecuenciaDeCollatz sopa4 5</code>	[5,16]
<code>mayorSecuenciaDeCollatzPermutando sopa4 5</code>	[5,16,8,4,2,1]
<code>mayorSecuenciaDeCollatzPermutando sopa1 32</code>	[32,16,8,4,2]
<code>mayorSecuenciaDeCollatzPermutando sopa1 8</code>	[8,4,2]
<code>mayorSecuenciaDeCollatzPermutando sopa2 4</code>	[4,2,1]
<code>mayorSecuenciaDeCollatzPermutando sopa3 11</code>	[11,34,17]
<code>mayorSecuenciaDeCollatzPermutando sopa2 1</code>	[1]
<code>mayorSecuenciaDeCollatzPermutando sopa3 2</code>	[]
<code>mayorSecuenciaDeCollatzPermutando sopa1 16</code>	[16,8,4,2]

Pautas de Entrega

- El trabajo se debe realizar en grupos de **3 alumnos**.
- El archivo con el código fuente debe tener nombre `nrogrupo_alg1-tp.hs` (por ej. para el grupo 3 el nombre debe ser `3_alg1-tp.hs`), y debe entregarse llenando el formulario que se encuentra en <http://bit.do/tp-algebra-v2020>. Además, en el archivo entregado debe indicarse, en un comentario arriba de todo: nombre y LU (o DNI) de cada integrante.
- El código debe poder ser ejecutado en el GHCI instalado en los laboratorios del DC, sin ningún paquete especial.
- No está permitido alterar los tipos de datos presentados en el enunciado, ni utilizar técnicas no vistas en clase para resolver los ejercicios (como por ejemplo, alto orden).

- Pueden definirse todas las funciones auxiliares que se requieran. Cada una debe tener un comentario indicando claramente qué hace.
- No es necesario entregar un informe sobre el trabajo.
- La fecha límite de entrega es el domingo 8 de marzo a las 23:59 hs.
- Se admitirá un único envío, sin excepción, por grupo.

Los objetivos a evaluar en el código de este trabajo práctico son:

- Resolución correcta.
- Declaratividad.
- Prolijidad: evitar repetir código innecesariamente y usar adecuadamente las funciones previamente definidas (tener en cuenta tanto las funciones definidas en el enunciado como las definidas por ustedes mismos).
- Uso adecuado de las técnicas vistas en clase como recursión o pattern matching. No está permitido resolver el trabajo con técnicas de programación no vistas en clase (por ejemplo, alto orden).

Referencias del lenguaje Haskell

- **The Haskell 2010 Language Report:** la última versión oficial del lenguaje Haskell a la fecha, disponible online en <http://www.haskell.org/onlinereport/haskell2010>.
- **Learn You a Haskell for Great Good!:** libro accesible, para todas las edades, cubriendo todos los aspectos del lenguaje, notoriamente ilustrado, disponible online en <http://learnyouahaskell.com> y en <http://aprendehaskell.es/> (en español).
- **Real World Haskell:** libro apuntado a zanjar la brecha de aplicación de Haskell, enfocándose principalmente en la utilización de estructuras de datos funcionales en la “vida real”, disponible online en <http://book.realworldhaskell.org/read>.
- **Hoogle:** buscador que acepta tanto nombres de funciones y módulos, como signatures y tipos *parciales*, online en <http://www.haskell.org/hoogle/>