



Práctica 6 : Secuencias

1. Usando la función `scan` y otras funciones del TAD Secuencias, definir las siguientes funciones:

- a) `promedios : Seq Int → Seq Float`, que dada una secuencia de enteros calcule el promedio de cada comienzo de la lista. Por ejemplo,

$$\text{promedios}\langle 2, 4, 3, 7, 9 \rangle = \langle 2, 3, 3, 4, 5 \rangle$$

- b) `mayores : Seq Int → Int`, que dada una secuencia de enteros devuelva la cantidad de enteros en la secuencia que son mayores a todos los anteriores. Por ejemplo,

$$\text{mayores}\langle 1, 2, 5, 3, 5, 2, 7, 9 \rangle = 4$$

2. Los números de Fibonacci son una secuencia de enteros dados por la siguiente recurrencia¹:

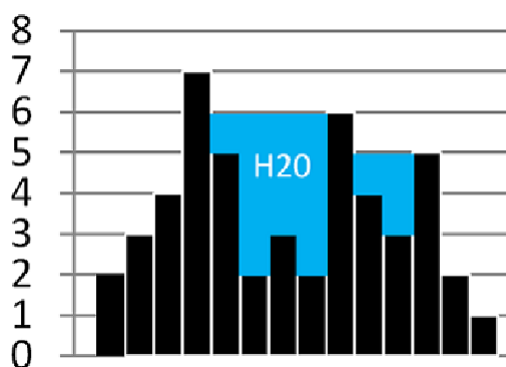
$$\begin{aligned} F_{-1} &= 1 \\ F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{aligned}$$

Usando esta definición es posible probar por inducción la siguiente propiedad:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

Dar una definición de la función `fibSeq : Nat → Seq Nat`, que dado un natural n calcule la secuencia de los primeros n números de Fibonacci, cuyo trabajo y profundidad sean $W(n) = n$ y $S(n) = \lg(n)$. Utilizar la función `scan`.

3. Considerar el siguiente problema: Si se vierte agua sobre un histograma, ¿cuánta agua (en términos del área) queda almacenada sobre el mismo? El siguiente dibujo puede servir como ilustración del problema dado.



El problema puede resolverse calculando la cantidad de agua que queda almacenada sobre cada barra del histograma. Por ejemplo, para una barra b_i la cantidad de agua que se acumula sobre ésta es igual al máximo entre 0 y

$$\min(\max L, \max R) - \text{altura}(b_i)$$

donde $\max L$ es el valor máximo entre las alturas de las barras que están a la izquierda de b_i y $\max R$ es el valor máximo entre las alturas de las barras que están a la derecha de b_i . Para el histograma de la figura el resultado sería 15.

¹La definición del elemento (-1)-ésimo de la secuencia de Fibonacci fue necesaria para que la propiedad dada sea cierta. La elección de este número no afecta los demás números de la secuencia.

Definir una función `aguaHist` que dada una secuencia de enteros (que represente un histograma) devuelva la cantidad de agua almacenada, utilizando las funciones `scan` y `reduce`.

4. El problema de determinar si en una expresión los paréntesis están bien anidados (todo paréntesis que se abre se cierra, y en ningún prefijo hay más cerrados que abiertos), puede resolverse utilizando una función `matchParen : Seq Paren → Bool`, que dada una secuencia del tipo:

data Paren = Open | Close

(que representa los caracteres '(' y ')'), devuelva `True` si la secuencia contiene la misma cantidad de valores `Open` que `Close` y además cada prefijo de la secuencia no contiene menos valores `Open` que `Close`, y `False` en caso contrario.

- a) Dar una definición de `matchParen` que implemente un algoritmo “*Divide & Conquer*”. La función `showT` del TAD de secuencias puede ser de utilidad para dividir una secuencia en dos.

Ayuda: Definir una función `matchP : Seq Paren → (Int, Int)` tal que si `matchP s = (i, j)` entonces `s` puede reducirse a `)i(j` eliminando todas las subsecuencias de la forma `()`. La función `matchParen` quedaría definida en términos de esta función como:

`matchParen s = matchP s ≡ (0, 0)`

- b) Dar una definición de `matchParen` que utilice la función `scan`. Comparar los costos (trabajo y profundidad) de las dos versiones.

5. Dada una secuencia de enteros, el problema conocido como “la subsecuencia contigua creciente más larga” consiste en encontrar la cantidad mayor de crecimientos contiguos en una secuencia.

Por ejemplo,

`sccml <9, 3, 5, 1, 3, 4, 5, 6, 8, 1> = 5`
`sccml <5, 6, 2, 3, 5, 1, 9> = 2`
`sccml <1, 4, 6, 7, 8, 11, 12, 3> = 6`

- a) Definir una función `sccml : Seq Int → Int`, que resuelva el problema utilizando un algoritmo “*Divide & Conquer*”. Describir qué valores deben calcular las llamadas recursivas para una eficiente construcción de la solución.

- b) Dar otra definición de la función `sccml`, utilizando la función `scan`.

6. Dada una secuencia de enteros `s`, se desea determinar el tamaño del siguiente conjunto:

$$\text{multiplos}(s) = \{(i, j) \mid 0 \leq i < j < |s|, \text{mod}(s_i, s_j) == 0\}$$

Por ejemplo, `multiplos (<12, 4, 6, 3, 2>)` tiene tamaño 7, `multiplos (<4, 6, 2>)` 2 y `multiplos (<1, 2, 3, 4, 5>)` 0.

Dar una definición de la función `cantMultiplos : Seq Int → Int`, que dada una secuencia `s` calcule el tamaño de `Multiplos (s)`, en término de operaciones paralelizables, como por ejemplo `map` y `reduce`.

7. Definir las siguientes operaciones que permiten extender el TAD Secuencias.

- a) `merge : (a → a → Ordering) → Seq a → Seq a → Seq a`, que dadas una relación de orden y dos secuencias ordenadas (respecto a ésta relación) `s1` y `s2`, construye una secuencia ordenada con los elementos de `s1` y `s2`. Definir la función utilizando un algoritmo “*Divide & Conquer*”.

- b) `sort : (a → a → Ordering) → Seq a → Seq a`, que ordena una secuencia según una relación de orden dada.

- c) `maxE : (a → a → Ordering) → Seq a → a`, que devuelve el máximo de una secuencia.

- d) `maxS : (a → a → Ordering) → Seq a → Nat`, que devuelve el índice de un máximo en la secuencia.

- e) `group : (a → a → Ordering) → Seq a → Seq a`, que dada una secuencia agrupa los elementos iguales contiguos. Por ejemplo,

`group <1, 1, 2, 3, 4, 4, 2, 2> = <1, 2, 3, 4, 2>`

- f) $\text{collect} : \text{Seq } (a, b) \rightarrow \text{Seq } (a, \text{Seq } b)$, que recolecta todos los datos asociados a cada clave y devuelve una secuencia de pares ordenada según el primer elemento. Por ejemplo,

$$\text{collect } \langle (2, "A"), (1, "B"), (1, "C"), (2, "D") \rangle = \langle (1, \langle "B", "C" \rangle), (2, \langle "A", "D" \rangle) \rangle$$

8. Una universidad cuenta con una base de datos de los estudiantes que rindieron los exámenes de ingreso y las notas de las evaluaciones realizadas. Un estudiante ingresará la universidad si el promedio de los resultados de los exámenes es mayor o igual a 70 y quedará en lista de espera si el promedio es mayor a 50 y menor a 70. Se desea saber cuántos estudiantes aprobaron el ingreso, cuántos quedaron en lista de espera y cuántos no podrán ingresar a la universidad, junto con las notas máxima de cada uno de los 3 casos.

Definir una función $\text{datosIngreso} : \text{Seq } (\text{String}, \text{Seq } \text{Int}) \rightarrow \text{Seq } (\text{Int}, \text{Int})$ que dada una secuencia con los nombres de los estudiantes y las notas de los exámenes, calcule los datos necesarios (cantidad de alumnos, nota máxima) de los exámenes de ingreso. Definirla en términos de `mapCollectReduce`.

9.

- a) Definir una función $\text{countCaract} : \text{Seq } (\text{Seq } \text{Char}) \rightarrow \text{Seq } (\text{Char}, \text{Int})$ que dada una colección de textos calcule la cantidad de veces con que aparecen los caracteres en los textos. Definirla en términos de `mapCollectReduce`.
- b) Definir una función $\text{huffman} : \text{Seq } (\text{Seq } \text{Char}) \rightarrow \text{Seq } (\text{Int}, \text{Seq } \text{Char})$ que dada una colección de textos calcule las frecuencias con que cada caracter aparece en los textos. La secuencia resultado debe contener pares de la forma $(n, \text{caracteres con frecuencia } n)$ y debe estar ordenada según las frecuencias de los caracteres. Utilizar las funciones `countCaract`, `map` y `collect`.