

Problemario de Semana IV: Haskell – Orden Superior, Tipos y Entrada/Salida

1. Escriba las siguientes funciones en Haskell, utilizando recursión de cola.

a) `factorial :: Integer -> Integer`

Debe calcular el factorial del entero proporcionado.

b) `log2Iter :: Integer -> Integer`

Debe calcular el logaritmo entero iterado (en base 2) del entero proporcionado.

c) `reverso :: [a] -> [a]`

Debe calcular el reverso de la lista suministrada.

d) `replicar :: Integer -> a -> [a]`

Debe devolver una lista con el elemento suministrado, repetido tantas veces como se especifique.
(por ejemplo: `replicar 3 True = [True, True, True]`).

e) `mezclarCon :: (a -> b -> c) -> [a] -> [b] -> [c]`

Debe tomar dos listas y operar los elementos de cada una, que estén en la misma posición, con la función propuesta.

2. Escriba las siguientes funciones en Haskell, utilizando una llamada (no trivial) a `foldr` o `foldl`.

a) `suma :: (Num a) => [a] -> a`

Debe calcular la suma de la lista proporcionada.

b) `filtra :: (a -> Bool) -> [a] -> [a]`

Debe calcular la lista que resulta de filtrar los elementos que no cumplen con la función booleana proporcionada.

c) `ordenar :: (Ord a) => [a] -> [a]`

Debe calcular la lista que resulta de ordenar los elementos de la lista proporcionada, de menor a mayor.

3. Defina el tipo de datos `ArbolTernario a`, con constructores:

■ `Hoja a`

■ `Rama (Arbol a) a (Arbol a) a (Arbol a)`

, e implemente luego las siguientes funciones.

a) `aplanar :: Arbol a -> [a]`

Debe devolver un recorrido inorder del árbol.

b) `esDeBusqueda :: (Ord a) => Arbol a -> Bool`

Debe ver si el árbol ternario en cuestión es de búsqueda o no.

c) `mapear :: (a -> b) -> Arbol a -> Arbol b`

Debe aplicar la función dada a todos los elementos de un árbol.