

Práctico 3: Programación Funcional

El objetivo de estos ejercicios es ayudar a afianzar los conocimientos sobre programación funcional. En cada ejercicio se debe dar el perfil de la función

NOTA Los ejercicios con * son para resolver en su casa

1. Defina una función que, dadas dos listas **ys** y **xs** de naturales ordenadas, retorne el *merge* de estas listas, es decir, la lista ordenada compuesta por los elementos de **ys** y **xs**.
2. Defina una función que, dada una lista de naturales, la ordene.
3. Defina una función que, recursivamente y sólo utilizando adición y multiplicación, calcule, dado un natural n , el número 2^n .
- 4 *. Defina una función que, dado un número natural n , retorne su representación binaria como secuencia de bits.
5. Defina una función que, dado un número natural n en su representación binaria, decida si n es par o no.
6. Escriba una función que, dado un número natural, decida si el mismo es un cuadrado perfecto o no.
7. Definir la función *repetir* de forma tal que *repetir* n x es una lista con n copias del elemento x . Dar dos definiciones, una de ellas con listas por comprensión.
8. Redefinir la función *minimum* tal que *minimum* l es el menor elemento de la lista. Dar dos definiciones, una de ellas utilizando funciones de alto orden (*foldl1*).

Árbol binario

Dada la definición del tipo genérico de un árbol binario:

Data Arbol a = Vacio | Nodo (Arbol a) a (Arbol a) deriving Show

Consideramos las tres componentes del constructor *Nodo* son el subárbol izquierdo, la raíz y el subárbol derecho respectivamente.

Definir las siguientes funciones:

9. *Pertenece*, que verifique si un elemento pertenece a un árbol binario.

10.Profundidad, que calcule la profundidad de un árbol binario.

11.Tamaño, que calcule la cantidad de nodos de un árbol binario.

11.Balanceado, se dice que un árbol es balanceado si para cada nodo se tiene que el número de hojas en cada uno de sus subárboles difiere como máximo en uno y sus subárboles son balanceados. Definir la función