Exercises
Declaring Types, Trees

1. Consider the following type of binary trees:

   ```
   data Tree a = Leaf a | Node (Tree a) (Tree a)
   ```

   Let us say that such a tree is *balanced* if the number of leaves in the left and right subtree differs by at most one, with the leaves themselves being trivially balanced.

   (a) Define a function ***size*** that returns the number of leaves in a tree.

   ```
   size:: Tree a -> Int
   ```

   (b) Using ***size*** above, or otherwise, define a funcion ***balanced*** that decides if a tree is balanced or not.

   ```
   balanced:: Tree a -> Bool
   ```

2. Define a function

   ```
   balance :: [a] -> Tree a
   ```

   that converts a non-empty list into a balanced tree.
   Hint: first define a function that splits a list into two halves whose length differs by at most one.