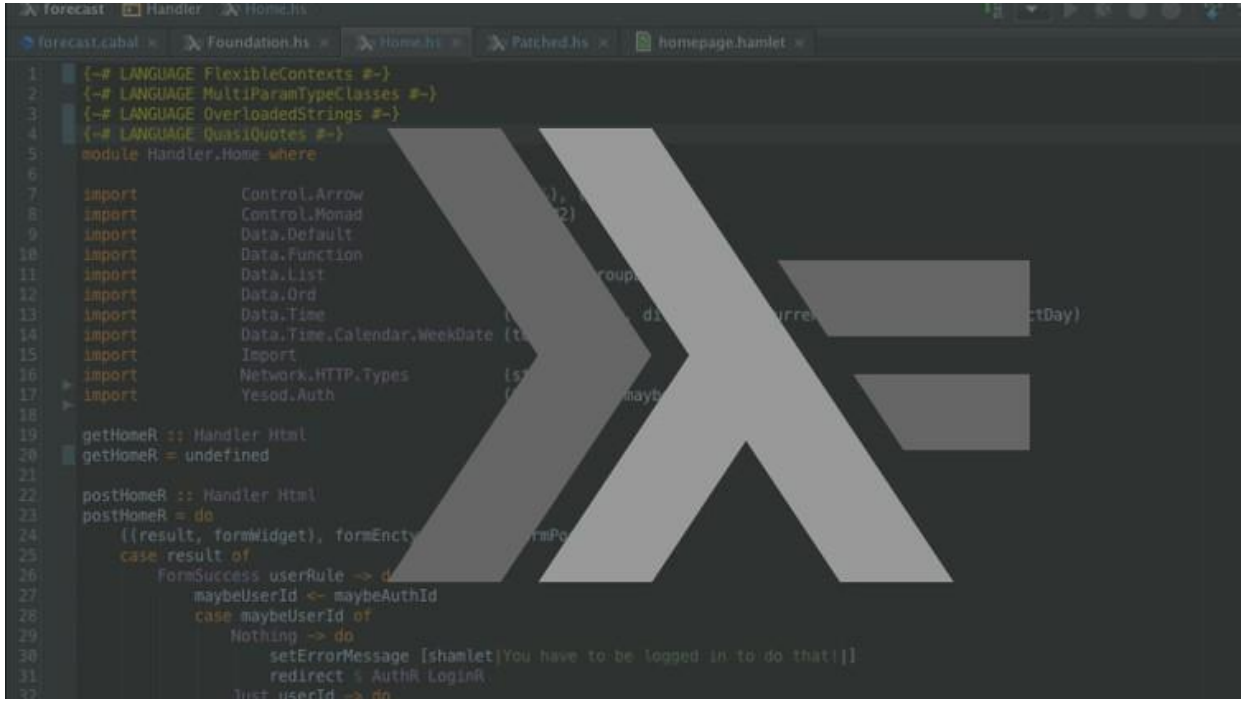
Binary Trees Problems



Binary Trees Problems

The definition of the trees is given by:

data *Tree a = Node a (Tree a) (Tree a) | Empty* **deriving (Show)**

That is, a tree with elements of type *a* is, either an empty tree, either a node with an element (of type *a*) and two other trees of the same type. The *deriving (Show)* statement simply enables an visualization of trees.

Problem 8

Write a function *`breadthFirst :: Tree a -> [a]`* that, given a tree, return its traversal by levels.

Public test cases

Input		Output

`let t7 = Node 7 Empty Empty`
`let t6 = Node 6 Empty Empty`
`let t5 = Node 5 Empty Empty`
`let t4 = Node 4 Empty Empty`
`let t3 = Node 3 t6 t7`
`let t2 = Node 2 t4 t5`
`let t1 = Node 1 t2 t3`
`let t1' = Node 1 t3 t2`
	`breadthFirst t1`	`[1,2,3,4,5,6,7]`

Breadth First Search

It starts at the tree root and explores **all nodes** at the present depth prior to moving on to the nodes at the next depth level.

Instructor Youtube Channel: Lucas Science