Lösungen zu Haskell

Aufgabe 1

```haskell
--Aufgabe 1.1
data Tree a = Branch (Tree a) a (Tree a) | Leaf deriving Show

--Aufgabe 1.2
search :: (Eq a) => Tree a -> a -> Bool
search Leaf _ = False
search (Branch t1 y t2) x = x == y || search t1 x || search t2 x

--Aufgabe 1.3
search' :: (Ord a) => Tree a -> a -> Bool
search' Leaf _ = False
search' (Branch t1 y t2) x | x == y = True
    | x < y = search t1 x
    | otherwise = search t2 x

--Aufgabe 1.4
insert :: (Ord a) => a -> Tree a -> Tree a
insert x Leaf = Branch Leaf x Leaf
insert x (Branch t1 y t2) | x <= y = Branch (insert x t1) y t2
    | otherwise = Branch t1 y (insert x t2)

--Aufgabe 1.5
list2Tree :: (Ord a) => [a] -> Tree a
list2Tree = foldr insert Leaf

--Aufgabe 1.6
tree2List :: (Ord a) => Tree a -> [a]
tree2List Leaf = []
tree2List (Branch t1 x t2) = (tree2List t1) ++ (x:tree2List t2)

--Aufgabe 1.7
quicksort :: (Ord a) => [a] -> [a]
quicksort = tree2List . list2Tree
```