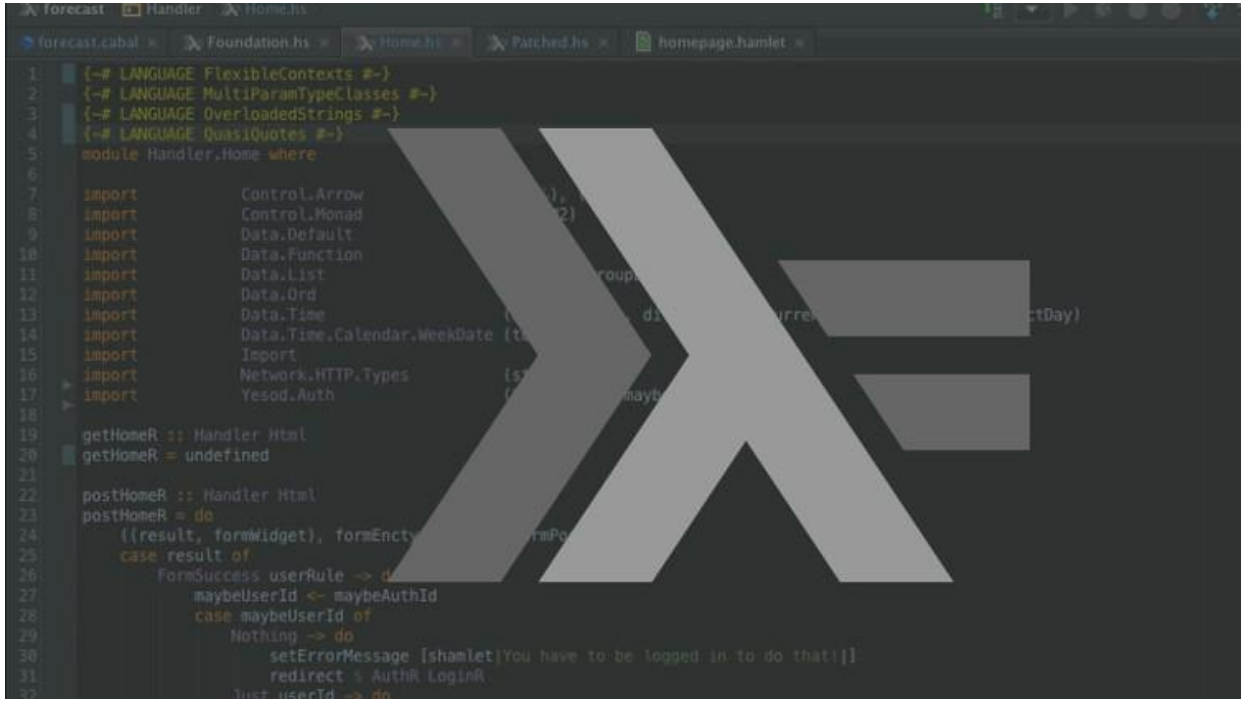
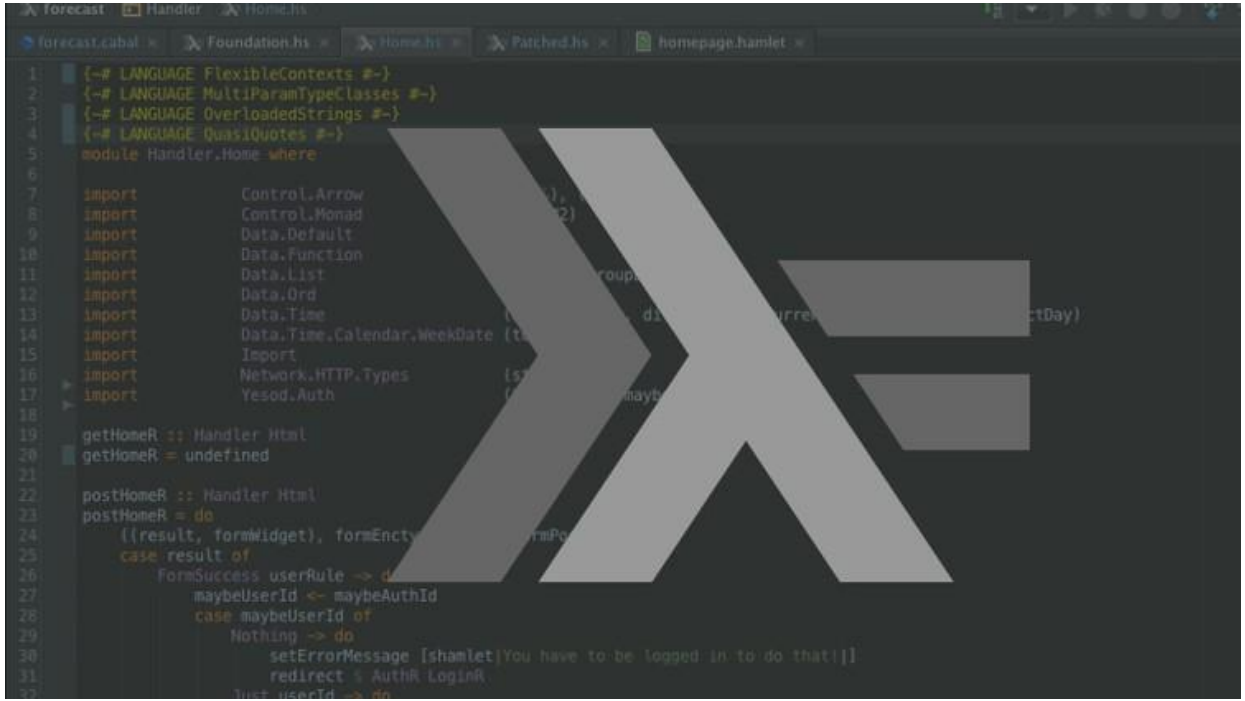
Multiway Tree Problems



Problem 4

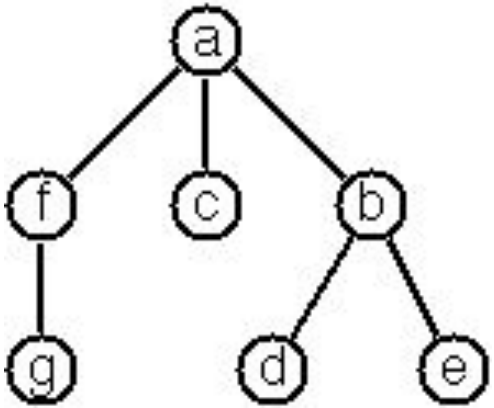

forecast.cabal
Foundation.hs
Home.hs
Patched.hs
homepage.hamlet
{-# LANGUAGE FlexibleContexts #-}
{-# LANGUAGE MultiParamTypeClasses #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE QuasiQuotes #-}
module Handler.Home where
import Control.Arrow
import Control.Monad
import Data.Default
import Data.Function
import Data.List
import Data.Ord
import Data.Time
import Data.Time.Calendar.WeekDate
import Import
import Network.HTTP.Types
import Yesod.Auth
getHomeR :: Handler Html
getHomeR = undefined
postHomeR :: Handler Html
postHomeR = do
((result, formWidget), formEncty
case result of
FormSuccess userRule -> d
maybeUserId <- maybeAuthId
case maybeUserId of
Nothing -> do
setErrorMessage [shamlet|You have to be logged in to do that!|]
redirect $ AuthR LoginR

Problem 4

Write a function *bottomUp :: Tree a -> [a]* that constructs the bottom-up sequence of the nodes of the multiway tree.

Example below:

```
bottomUp = "gfcdeba"
```



Problem 4

Examples:

```
bottomUp (Node 'a' [Node 'f' [Node 'g' []],Node 'c' [],Node 'b' [Node 'd' [], Node 'e' []]])
```

-> **"gfcdeba"**

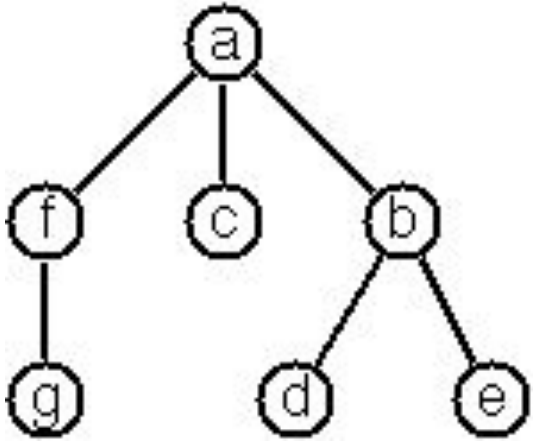
```
bottomUp (Node 'a' [Node 'b' [Node 'c' []]])
```

-> **“cba”**

Haskell Multiway Trees

In Haskell, we define multiway trees as a datatype:

```haskell
data Tree a = Node a [Tree a]
        deriving (Eq, Show)
```

Example

```
tree1 = Node 'a' [
                  Node 'f' [Node 'g' []],
                  Node 'c' [],
                  Node 'b' [Node 'd' [], Node 'e' []]
                  ]
```