

Exercícios sobre Tipos Algébricos Recursivos

prof. André Rauber Du Bois

Universidade Federal de Pelotas
<http://sites.google.com/site/haskellufpel/>
dubois@inf.ufpel.edu.br

1 Questionário

Todos os exercícios devem usar a seguinte definição de árvore:

```
data Arvore a = Folha a | Nodo a (Arvore a) (Arvore a)
deriving(Eq, Show)
```

1. Defina a função `multDois`, que recebe uma árvore de inteiros e multiplica por 2 todos os inteiros guardados na árvore:

```
multDois :: Arvore Int -> Arvore Int
```

2. Defina a função

```
contaElementos :: Arvore a -> Int
```

que conta quantos elementos estão guardados na árvore

3. Defina a função `altura`, que calcula altura (quantos níveis uma árvore possui) de uma árvore qualquer.

4. Defina a função `maiorElemento` que acha o maior elemento de uma árvore de inteiros

5. Defina a função

```
procuraInt :: Int -> Arvore Int -> Bool
```

que diz se um inteiro ocorre dentro de uma árvore de inteiros

6. Defina a função

```
quantasVezes :: Int -> Arvore Int -> Int
```

que conta quantas vezes um inteiro ocorre dentro de uma árvore de inteiros

7. Uma árvore refletida é uma árvore com seus ramos esquerdos e direitos trocados. Defina a função

```
refleteArvore :: Arvore a -> Arvore a
```

8. Defina a função

```
ArvoreToLista :: Arvore a -> [a]
```

que transforma uma árvore em uma lista

9. Defina a função `mapTree` que aplica uma função a todos os valores guardados em uma árvore.

```
mapTree :: (a -> b) -> Arvore a -> Arvore b
```