Exercícios sobre Listas II

prof. André Rauber Du Bois

Universidade Federal de Pelotas
dubois@inf.ufpel.edu.br

1 Questionário

1. Defina a função

   ```
   membro :: Int -> [Int] -> Bool
   ```

   que retorna um booleando que diz se um inteiro está presente na lista
2. Implemente a função

   ```
   membroNum ::  Int -> [Int] -> Int
   ```

   que conta o número de vezes que um inteiro aparece em uma lista
3. Defina a função `membro` usando a função `membroNum`
4. Implemente a função

   ```
   unico :: [Int] -> [Int]
   ```

   que retorna uma lista com os números que aparecem apenas uma vez na lista argumento. Ex:

   ```
   *Main> unico [2,4,1,4,1,3]
   [2,3]
   ```

   A função `memberNum` deve ser usada na definição de `unico`.
5. Observe a implementação da função de ordenação quickSort:

   ```
   quickSort :: [Int] -> [Int]
   quickSort [] = []
   quickSort (x:xs) = quickSort (menores x xs)
                      ++ [x] ++
                      quickSort (maiores x xs)
   ```

   A intuição dessa solução é a seguinte: deixar o primeiro elemento da lista (pivô) no meio, então colocar antes dele todos os elementos menores que o pivô ordenados e, depois dele, todos os elementos mairoes que o pivô ordenados. Implemente as funções

   ```
   menores :: Int -> [Int] -> [Int]
   maiores :: Int -> [Int] -> [Int]
   ```

   e veja o quick sort funcionando!