

Lista de exercícios - Recursão

Recursão primitiva sobre os naturais

1. Dada uma função f de Int em Int , defina por recursão primitiva uma função maiorF que aceite um natural n e devolva o maior valor dentre $f\ 0, f\ 1, \dots$ e $f\ n$.

Para testar ou validar, use a seguinte definição para f

```
f :: Int -> Int
f m
  | m == 0 = 8
  | m == 1 = 44
  | m == 2 = 17
  | otherwise = 0
```

2. Dada uma função f de Int em Int , defina por recursão primitiva uma função algumF0 que aceite um natural n e devolva o booleano `True` se e somente se um ou mais valores dentre $f\ 0, f\ 1, \dots, f\ n$ é zero. Teste com diferentes definições de f .
3. Dada uma função f de Int em Bool , defina por recursão primitiva uma função algumFentre que aceite um natural n e devolva o booleano `True` se e somente se $f\ i$ é `True` para algum i entre 0 e n . Teste com diferentes definições de f .
4. Defina por recursão primitiva uma função que calcule a raiz quadrada inteira de n (o maior natural cujo quadrado é menor ou igual a n).

Recursão primitiva sobre listas

5. Sem usar compreensões, defina uma função que dada uma lista xs e um valor x calcule a lista obtida de xs eliminado a primeira ocorrência de x , caso ele ocorrer em xs , senão retornar o próprio xs .
6. Sem usar compreensões, defina uma função para eliminar todas as ocorrências de um dado elemento.
7. Sem usar compreensões, defina uma função que receba um `String` e retorne o correspondente `String` invertido.
8. Defina uma função
 `unique :: [Int] -> [Int]`
tal que `unique xs` retorne a lista dos elementos de xs os quais ocorrem exatamente uma vez. Por exemplo, `unique [4,2,1,3,2,3]` é `[4,1]`. Pense em duas soluções, uma usando compreensões e outra não.

9. Altere a definição da função `ins`, dada em sala de aula, de tal forma que a função `iSort` retorne a lista ordenada em ordem decrescente.
10. Altere a definição da função `ins`, dada em sala de aula, de tal forma que a função `iSort` retorne a lista ordenada em ordem ascendente e ao mesmo tempo elimine elementos duplicados. Por exemplo, com esta definição deveremos ter que `iSort [2,1,4,1,23] = [1,2,4]`.

Recursão Geral

11. Defina uma função para calcular o menor de uma lista de inteiros.
12. Uma propriedade do `mdc` (máximo divisor comum) é que se $a > b$, então

$$\text{mdc } a \ b = \text{mdc } (a-b) \ b$$
 Utilize esta propriedade para dar uma definição recursiva para a função `mdc`.

13. Defina uma função


```
divMod :: Int -> Int -> (Int, Int)
```

 que calcula ao mesmo tempo a divisão inteira e o resto da divisão inteira. Assuma que sempre os argumentos serão inteiros positivos.

14. Considere a função

```
pot :: Int -> Int -> Int
pot m n
  | n == 0      = 1
  | otherwise = m * pot m (n-1)
```

Considerando $n > 0$, quantas chamadas recursivas são feitas para calcular `pot m n`? Podemos melhorar muito a eficiência de `pot` usando os seguintes fatos

- $m^n = (m^k)^2$, se n é par com $n = 2k$
- $m^n = m(m^k)^2$, se n é ímpar com $n = 2k + 1$

Dê uma definição para `pot` que se baseie nestes fatos. Nesta versão, quantas chamadas recursivas são feitas para calcular `pot m n`?

15.
 - Modifique a função `qSort` vista em sala de aula de tal forma que ordene os elementos decrescentemente.
 - Adapte a função `qSort` para que além de ordenar, elimine elementos duplicados.
16. Uma lista é uma **sublista** de outra se os elementos da primeira ocorrem na segunda, na mesma ordem. Por exemplo, "ship" é uma sublista de "Fish & Chips", mas não de "hippies". Defina uma função que decida se uma lista é uma sublista de outra.
17. Uma lista é uma **subsequência** de outra se ela ocorre como uma sequência seguida (um exatamente depois do outro) de elementos. Por exemplo, "Chip" é uma subsequência de "Fish & Chips", mas não de "Chin up". Defina uma função que decida se uma lista é uma subsequência de outra.

Desafio

18. Ana descobriu um livro cujo título dizia
EHT CDOORRSSW AAAGMNR ACDIINORTY
O conteúdo tem uma sequência de entradas como esta

```
6-letter words
-----
...
eginor: ignore, region
eginrr: ringer
eginrs: resign, signer, singer
...
```

Ana descobriu que o livro é um dicionário de anagramas. As letras dos anagramas estão ordenadas e os resultados estão listados seguindo a ordem do dicionário. Associado com cada anagrama estão as palavras, em Inglês, com as mesmas letras. Defina uma função

```
anagrams :: Int -> [String] -> String
```

tal que `anagrams` receba como argumentos um inteiro positivo `n` e uma lista `ws` de palavras do Inglês ordenadas alfabeticamente, e retorne um string que contenha uma lista de entradas de anagramas para todas as palavras de longitude `n`.