

IF686 - Paradigmas de Linguagens Computacionais

Primeira lista de exercícios — Haskell

1. (0,2 points) Usando recursão, implemente a função `sumTo`, de modo que `sumTo n` calcule o valor de $1 + 2 + \dots + n$
2. (0,2 points) Defina a função `potencia`, de modo que `potencia n k` calcule n elevado a k . Use recursão.

3. (0,2 points) Usando recursão, compute os coeficientes binomiais dados pelas seguintes equações

$$B(n, k) = B(n - 1, k) + B(n - 1, k - 1)$$

$$B(n, 0) = 1$$

$$B(0, k) = 0, \text{ quando } k > 0$$

Dica: usar casamento de padrão pode ser de grande ajuda.

4. (0,2 points) Os números de Tribonacci são dados pelas seguintes equações

$$T(1) = 1$$

$$T(2) = 1$$

$$T(3) = 2$$

$$T(n + 1) = T(n) + T(n - 1) + T(n - 2)$$

Implemente uma função recursiva eficiente que calcule `T n`. Considere o uso de uma função auxiliar que leva em consideração o conceito de recursão de cauda.

5. (0,2 points) Defina a função

`squares :: Int -> [Integer]`

que encontra os n primeiros quadrados que começam e terminam com o mesmo dígito. Por exemplo,

`squares 9 ~> [1,4,9,121,484,676,1521,1681,4624]`