

Algoritmos e recursão

4.1 Escreva uma definição da função `algarismos :: Int → [Int]` que obtém os algarismos decimais de um inteiro positivo. Exemplo: `algarismos 12345 = [1, 2, 3, 4, 5]`.

Sugestão: Pode obter o algarismo das unidades usando o resto da divisão por 10 e prosseguir recursivamente com o quociente da divisão. Comece por definir uma função auxiliar que obtenha os algarismos pela ordem inversa, i.e. `algarismosRev 12345 = [5, 4, 3, 2, 1]`.

4.2 Escreva uma definição da função `toBits :: Int → [Int]` que obtém a representação em binário de um inteiro não-negativo. Exemplo: `toBits 29 = [1, 1, 1, 0, 1]`. Note que os dígitos binários do resultado estão pela ordem do mais significativo para o menos significativo.

Sugestão: O problema é semelhante ao exercício anterior, mas efetuando divisões por 2 em vez de 10.

4.3 Escreva uma definição da função `fromBits :: [Int] → Int` que faz a transformação inversa da anterior, ou seja, converte dígitos em binário para o inteiro não-negativo correspondente.

4.4 O algoritmo de Euclides para calcular o máximo divisor comum de dois inteiros a, b pode ser expresso de forma recursiva:

$$\text{mdc}(a, b) = \begin{cases} a, & \text{se } b = 0 \\ \text{mdc}(b, a \bmod b), & \text{caso contrário} \end{cases}$$

Traduza esta definição recursiva para uma função `mdc :: Integer → Integer → Integer`.

4.5 Ordenação de listas pelo **método de inserção**.

- (a) Escreva definição recursiva da função `insert :: Ord a ⇒ a → [a] → [a]` da biblioteca `List` para inserir um elemento numa lista ordenada na posição correcta de forma a manter a ordenação. Exemplo: `insert 2 [0, 1, 3, 5] = [0, 1, 2, 3, 5]`.
- (b) Usando a função `insert`, escreva uma definição também recursiva da função `isort :: Ord a ⇒ [a] → [a]` que implementa *ordenação pelo método de inserção*:
 - a lista vazia já está ordenada;
 - para ordenar uma lista não vazia, recursivamente ordenamos a cauda e inserimos a cabeça na posição correcta.

4.6 Ordenação de listas pelo **método de seleção**.

- (a) Escreva definição recursiva da função `minimum :: Ord a => [a] -> a` do prelúdio-padrão que calcula o menor valor numa lista não-vazia. Exemplo: `minimum [5, 1, 2, 1, 3] = 1`.
- (b) Escreva uma definição recursiva da função `delete :: Eq a => a -> [a] -> [a]` da biblioteca `List` que remove a primeira ocorrência dum valor numa lista. Exemplo: `delete 1 [5, 1, 2, 1, 3] = [5, 2, 1, 3]`.
- (c) Usando as funções anteriores, escreva uma definição recursiva da função `ssort :: Ord a => [a] -> [a]` que implementa *ordenação pelo método de seleção*:
 - a lista vazia já está ordenada;
 - para ordenar uma lista não vazia, colocamos à cabeça o menor elemento m e recursivamente ordenamos a cauda sem o elemento m .

4.7 Ordenação de listas pelo método **merge sort**.

- (a) Escreva uma definição recursiva da função `merge :: Ord a => [a] -> [a] -> [a]` para juntar duas listas ordenadas numa só mantendo a ordenação. Exemplo: `merge [3, 5, 7] [1, 2, 4, 6] = [1, 2, 3, 4, 5, 6, 7]`.
- (b) Usando a função `merge`, escreva uma definição recursiva da função `msort :: Ord a => [a] -> [a]` que implementa o método *merge sort*:
 - uma lista vazia ou com um só elemento já está ordenada;
 - para ordenar uma lista com dois ou mais elementos, partimos em duas metades, recursivamente ordenamos as duas partes e juntamos os resultados usando `merge`.

Sugestão: comece por definir uma função `metades :: [a] -> ([a], [a])` para partir uma lista em duas metades (ver a Folha 1).

4.8 (T) Podemos fazer operações com polinómios representados pelas listas de coeficientes. Por exemplo, podemos representar dois polinómios $P = 1 + 3X - X^2$ e $Q = 3 + 2X^2$ pelas listas $[1, 3, -1]$ e $[3, 0, 2]$, respectivamente.

- (a) Escreva uma definição da função `addPoly :: [Int] -> [Int] -> [Int]` que calcula a soma de dois polinómios. Exemplo: `addPoly [1, 3, -1] [3, 0, 2] = [4, 3, 1]` é a lista dos coeficientes de $P + Q$.
- (b) Escreva uma definição da função `multPoly :: [Int] -> [Int] -> [Int]` que calcula o produto de dois polinómios. Exemplo: `multPoly [1, 3, -1] [3, 0, 2] = [3, 9, -1, 6, -2]` é a lista de coeficientes de $P \times Q$.