

Problema C Exemplo - Programação Funcional 2023/2024

Algoritmia e Programação

Exercício 1

Utilize a técnica de memoização na implementação da função `tribonacci : int -> int` sobre inteiros naturais, traduzida pela expressão seguinte:

$$\text{tribonacci}(n) = \begin{cases} 1 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ 1 & \text{se } n = 2 \\ \sum_{i=1}^3 \text{tribonacci}(n-i) & \text{para } n > 2 \end{cases}$$

Resoluções sem um uso pertinente de memoização não serão contabilizadas.

Exercício 2

Utilizando pilhas, implemente a função `processa : char list -> char list -> bool` que indica se duas listas de *chars* representam a mesma palavra após processamento das mesmas. Ambas as listas apenas contêm letras em minúscula ou o carácter `#`, que representa o *backspace* (i.e., elimina a letra anterior).

Exemplo: `processa ['a';'b';'#';'c'] ['a';'d';'#';'c'] = true`.

Resoluções sem um uso pertinente de pilhas não serão contabilizadas.

Exercício 2 (Extra)

Utilizando pilhas, implemente a função `ordena : int list -> int list` que dada uma lista representada por $[a_0; a_1; \dots; a_{n-1}; a_n]$ retorne a lista reordenada da seguinte forma: $[a_0; a_n; a_1; a_{n-1}; a_2; a_{n-2}; \dots]$. Pode assumir que a lista original não tem números repetidos.

Exemplo: `ordena [1; 2; 3; 4; 5] = [1; 5; 2; 4; 3]`

Resoluções sem um uso pertinente de pilhas não serão contabilizadas.