

SCC0606 – Estruturas de Dados II

Prof. Dr. Renato Moraes Silva

Lista de Exercícios Práticos 2

Instruções gerais

Para cada exercício abaixo, crie uma pasta contendo:

- **um arquivo .h para a(s) função(ões) que resolve(m) o problema.** Nesse arquivo devem ser adicionadas definições de tipo, constantes, e protótipos das funções que fazem parte da sua interface pública.
- **um arquivo .c para a(s) função(ões) que resolve(m) o problema** com o mesmo nome do .h. Esse arquivo deve implementar as funções declaradas no arquivo .h.
- **um arquivo chamado main.c** que deve ser usado apenas para chamar e testar as funções que implementam o que foi pedido no exercício. Essa função deve chamar todos os arquivos .h necessários.
- Para compilar e executar o programa de forma automatizada, crie um arquivo **.bat** caso esteja utilizando Windows, ou um arquivo **.sh** se estiver utilizando Linux.

Exerc. 1. Defina a função recursiva `soma_naturais` que recebe como argumento um número natural n e devolve a soma de todos os números naturais até n .

Exemplo:

$$\text{soma_naturais}(5) = 5 + 4 + 3 + 2 + 1 = 15$$

Depois de fazer o código, analise qual é a complexidade assintótica de **tempo** e de **espaço** do algoritmo.

Exerc. 2. Implemente um algoritmo que utilize recursão para processar uma imagem representada por uma matriz bidimensional de valores em escala de cinza. O programa deve aplicar a operação de inversão de cores (negativo da imagem), transformando cada pixel p no seu complemento $255 - p$.

- A imagem será representada por uma matriz M de dimensões $N \times M$, onde cada elemento $M[i][j]$ assume valores entre 0 e 255.
- O algoritmo deve ser recursivo e percorrer a imagem pixel a pixel, aplicando a transformação $M[i][j] = 255 - M[i][j]$.

Veja um exemplo de entrada a seguir:

$$\begin{bmatrix} 10 & 50 & 100 \\ 150 & 200 & 250 \\ 30 & 70 & 120 \end{bmatrix}$$

Veja um exemplo de matriz que deveria ser retornada para a matriz de entrada apresentada acima:

$$\begin{bmatrix} 245 & 205 & 155 \\ 105 & 55 & 5 \\ 225 & 185 & 135 \end{bmatrix}$$

Depois de fazer o código, analise qual é a complexidade assintótica de **tempo** e de **espaço** do algoritmo.

Exerc. 3. Faça uma função recursiva que recebe como argumentos dois vetores w_1 e w_2 e devolve um vetor resultante que intercala os elementos de w_1 com os de w_2 . A intercalação deve ser feita pegando alternadamente um elemento de cada lista. Se uma das listas for maior, os elementos restantes devem ser adicionados ao final da lista resultante.

Exemplos:

- $\text{intercala}([1, 2, 3], [4, 5, 6]) \rightarrow [1, 4, 2, 5, 3, 6]$
- $\text{intercala}([1, 3, 5, 7, 9], [2, 4]) \rightarrow [1, 2, 3, 4, 5, 7, 9]$

Depois de fazer o código, analise qual é a complexidade assintótica de **tempo** e de **espaço** do algoritmo.