



UNIVERSIDADE FEDERAL DO CEARÁ  
Campus de Quixadá  
Prof. Arthur Araruna  
QXD0010- Estrutura de Dados

QXD0010  
**LE1**  
2023.1

Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_

**Observação:**

Você pode considerar a existência de um algoritmo chamado `GARANTIRCAPACIDADE`, que recebe uma Lista Sequencial e um natural representando uma capacidade mínima desejada, que se responsabiliza por realizar a realocação e a movimentação de elementos. (*Esse algoritmo faria basicamente o papel do código que fizemos em sala.*)

## Exercícios de Extensão

1. Escreva um algoritmo de remoção que, dada uma Lista Sequencial  $L$  e um valor de entrada  $x$ , remove de  $L$  todos os valores menores que  $x$ .
2. Refaça o algoritmo da Questão 1 de forma a garantir que ele passe pelos elementos do vetor apenas uma vez ao longo de todo o processo. (Em outras palavras, não usar mais que um laço).
3. Escreva uma versão de algoritmo de inserção que garanta não gerar elementos duplicados na Lista informada. Isso quer dizer que, caso o valor a inserir já exista, a inserção deve falhar.
4. Escreva um algoritmo para eliminar todas as ocorrências repetidas dos elementos em uma lista. Isso significa que, ao final da execução, cada elemento aparece uma única vez.
5. Podemos reescrever o algoritmo da Questão 4 de uma maneira mais eficiente caso a lista passada esteja ordenada? Justifique e, em caso afirmativo, escreva essa nova versão.
6. Considere que estivemos mantendo nossa Lista Sequencial  $L$  sempre disposta em *ordem crescente*. Para facilitar nosso trabalho, desejamos usar uma operação de inserção que garanta sempre preservar essa propriedade na nossa lista, independentemente do novo elemento a ser inserido. Escreva essa versão de inserção.

## Exercícios de Aplicação

7. Escreva um algoritmo que receba duas Listas Sequenciais  $L_1$  e  $L_2$ , que estarão garantidamente com seus elementos dispostos em *ordem crescente*, e remova de  $L_1$  quaisquer ocorrências dos elementos encontrados em  $L_2$ .
8. Refaça o algoritmo da Questão 7 de forma a garantir que ele passe pelos elementos de ambas as listas apenas uma vez ao longo de todo o processo. (De uma certa forma, não usar mais que um laço por lista).
9. Considere que estivemos usando Listas Sequenciais para representar *conjuntos*, de forma que precisamos garantir que nenhum elemento ocorra mais de uma vez em uma lista. Escreva algoritmos para representar as operações de `UNIÃO` e de `INTERSEÇÃO` entre duas dessas listas, de forma que o resultado seja retornado como uma nova lista.

## Desafios

10. Escreva um algoritmo que receba duas Listas Sequenciais, ambas com elementos dispostos em *ordem crescente*, e retorne uma nova lista contendo os elementos de ambas as listas de entrada, também dispostos em *ordem crescente*. Seu algoritmo deve funcionar sem a necessidade de alocar memória adicional (a não ser para a nova lista) e sem a necessidade de executar um procedimento de ordenação após a movimentação.