

# Estruturas de dados básicas II

## Lista de exercícios – Tipos abstratos de dados lineares

**Para cada questão abaixo, você deverá apresentar um algoritmo, analisar sua complexidade e implementá-lo.**

1. Implemente uma pilha que, além de permitir empilhar e desempilhar elementos, consegue retornar o elemento mínimo. Todas as operações deverão apresentar complexidade de pior caso constante.
2. Imagine uma pilha (literal) de pratos. Se a pilha ficar muito alta, ela poderá tombar. Assim, no mundo real, nós gostaríamos de começar uma nova pilha quando a pilha anterior excede um determinado limite de elementos. Implemente um TAD chamado *ConjuntoDePilhas* que possua este comportamento. Este TAD deverá ser composto de várias pilhas, criando uma nova pilha a cada vez que a pilha atual exceda um limite. No entanto, as funções de empilhar e desempilhar do seu TAD devem ser transparentes à quantidade de pilhas armazenadas internamente.
3. Implemente um algoritmo para solucionar o problema da Torre de Hanói sem recursão.
4. Implemente um algoritmo que leia uma sequência de caracteres e determine se os parênteses, colchetes e chaves presentes na sequência estão balanceados.
5. Implemente um algoritmo para remover chaves duplicadas de uma lista encadeada não-ordenada.
6. Uma lista encadeada pode ser implementada de diferentes formas. Escolha uma das formas estudadas em sala e apresente pseudocódigos e a análise de complexidade assintótica para um dos procedimentos abaixo:
  - (a) *swap*
  - (b) *merge*
  - (c) *reverse*
7. Expressões regulares são uma importante ferramenta para a verificação e extração de padrões em strings. Considere um modelo de expressão regular com os seguintes caracteres:

- a-z: letras minúsculas ASCII
- \*: uma ou mais ocorrências do caracter anterior
- .: uma ocorrência de qualquer caracter

Implemente um algoritmo que receba como entrada uma expressão regular bem formada e uma string, e avalie se a string informada pode ser descrita pelo padrão contido na expressão regular.

Exemplos de entrada/saída:

- “a”, “” : verdadeiro
  - “.”, “” : falso
  - “ab\*”, “a” : verdadeiro
  - “a.”, “ab” : verdadeiro
  - “a”, “a” : verdadeiro
8. Dado um vetor circular com N elementos binários, analisar qual sentido (horário ou anti-horário) apresenta a menor quantidade de inversões de bits (*bit flips*).

### Questões extras para quem quiser se aprofundar na área

1. A busca binária é uma abordagem de busca mais eficiente para o contexto de sequências. Nesta questão, você deverá apresentar uma busca binária recursiva adaptada para identificar o pico de uma *sequência bitônica*. Especificamente, uma sequência bitônica é uma sequência de  $n$  elementos

$$\{A_1, A_2, \dots, A_k, A_{k+1}, A_{k+2}, \dots, A_n\} \quad (1)$$

tal que:

- a subsequência  $\{A_1, A_2, \dots, A_k\}$  está em ordem crescente;
  - a subsequência  $\{A_k, A_{k+1}, \dots, A_n\}$  está em ordem decrescente.
  - o elemento  $A_k$  é dito o pico da sequência.
2. Implemente um algoritmo que ordene o conteúdo de uma pilha usando apenas uma pilha adicional.
3. Implemente:
- uma fila com duas pilhas. Qual a complexidade das operações de adicionar e remover elementos?
  - uma pilha com duas filas. Qual a complexidade das operações de empilhar e desempilhar?

4. Dada uma lista simplesmente encadeada, Implemente um algoritmo para encontrar o  $n$ -ésimo elemento, contado do fim da lista para o começo.
5. Dada uma lista simplesmente encadeada, Implemente um algoritmo para remover um nó, dado apenas acesso àquele nó.