



Lista de Exercícios Estruturas de Dados

prof^a Ana Carolina Sokolonski

20 de abril de 2026

1. Crie um struct Fila que utilize uma lista encadeada (ou um array, se preferir) para armazenar os elementos. O algoritmo deve implementar as seguintes funções:
 - (a) enfileirar(elemento): Adiciona um elemento ao final da fila.
 - (b) desenfileirar(): Remove e retorna o elemento do início da fila.
 - (c) espiar(): Retorna o elemento do início da fila sem removê-lo.
 - (d) esta_vazia(): Verifica se a fila está vazia.
 - (e) tamanho(): Retorna o número de elementos na fila.
2. Crie um algoritmo que utilize uma fila para verificar se uma palavra ou frase é um palíndromo (lê-se igual de frente para trás).
3. Simule o funcionamento de uma fila de impressão, onde tarefas são adicionadas à fila e processadas sequencialmente.
4. Implemente uma fila utilizando uma lista simplesmente encadeada circular para otimizar o uso de memória.

5. Crie uma classe Pilha que utilize uma lista ligada (ou um vetor, se preferir) para armazenar os elementos. A classe deve ter métodos para:
 - (a) empilhar(elemento): Adiciona um elemento ao topo da pilha.
 - (b) desempilhar(): Remove e retorna o elemento do topo da pilha.
 - (c) espiar(): Retorna o elemento do topo da pilha sem removê-lo.
 - (d) esta_vazia(): Verifica se a pilha está vazia.
 - (e) tamanho(): Retorna o número de elementos na pilha.
6. Crie um algoritmo que utilize uma pilha para reverter uma string.
7. Crie um algoritmo que utilize uma pilha para verificar se uma expressão matemática possui parênteses balanceados.
8. Implemente uma pilha utilizando um vetor.
9. Implemente uma árvore binária. A árvore deve ter métodos para:
 - (a) inserir(valor): Insere um novo nó na árvore.
 - (b) remover(valor): Remove um nó da árvore.
 - (c) buscar(valor): Busca um nó na árvore.
 - (d) percurso_em_ordem(): Realiza um percurso em ordem na árvore.
 - (e) percurso_pre_ordem(): Realiza um percurso pré-ordem na árvore.
 - (f) percurso_pos_ordem(): Realiza um percurso pós-ordem na árvore.
 - (g) calcular_altura(): Retorna a altura da árvore.
10. Implemente uma árvore de busca binária AVL. A árvore deve ter métodos para:
 - (a) inserir(valor): Insere um novo nó na árvore.
 - (b) remover(valor): Remove um nó da árvore.
 - (c) buscar(valor): Busca um nó na árvore.
 - (d) percurso_em_ordem(): Realiza um percurso em ordem na árvore.

- (e) `percurso_pre_ordem()`: Realiza um percurso pré-ordem na árvore.
 - (f) `percurso_pos_ordem()`: Realiza um percurso pós-ordem na árvore.
11. Crie uma árvore de expressão a partir de uma expressão matemática e utilize uma pilha para avaliar a expressão.