

SCC0502 - Algoritmos e estruturas de dados I

Prof. Dr. Bruno Matheus

Estagiário PAE: Rafael Nakanishi

Lista de Exercícios

Listas ordenadas e generalizadas

1 Listas Ordenadas

1. Implemente um TAD estático e um dinâmico para listas ordenadas contendo as seguintes funções principais: `cria()`, `insere()`, `remove()`, `limpa()` e `estaNaLista()`.
2. Implemente uma função que ordene uma lista (dinâmica e encadeada) de números não ordenados, trocando os ponteiros dos blocos.
3. Modifique a estrutura de Listas ordenadas para que ela seja circular. Implemente um TAD para essa estrutura.
4. Dados duas listas ordenadas, implemente uma função que retorne uma terceira lista ordenada que é a junção dos elementos das duas primeiras.
5. Dado uma lista ordenada, implemente uma função para alterar os elementos dessa lista. Ao final da modificação, a lista deve se manter ordenada.

2 Listas Generalizadas

1. Represente graficamente as seguintes listas generalizadas e responda quem são a cabeça e a lista de cada uma delas:
 - (a) $A = [a, [b, c]]$
 - (b) $B = [A, A]$
 - (c) $C = [[], a, [C]]$
 - (d) $D = [[a, b], [[c, d, []], [e]]]$
 - (e) $E = [[c, E, D, [e]], a, [b], cauda(E)]$

2. Dado uma estrutura de listas generalizadas, implemente as funções de um TAD para essa lista: `cria(.)`, `insereNoComeco(.)`, `insereNoFim(.)`, `remove(.)`, `limpa(.)`, `cabeca(.)` e `cauda(.)`.
3. Implemente uma função para listas generalizadas que verifica se um elemento está presente na lista (em qualquer lugar). Se estiver, retorna um ponteiro para a sublista do elemento encontrado, ou `NULL` caso contrário.
4. Supondo que todos os elementos da lista são únicos, implemente uma função que busque por um elemento e adicione um outro elemento na posição encontrada. Caso não encontre o elemento buscado, a inserção deve ser feita no final da lista. Por exemplo: ao inserir `g` na posição `d` da lista `[a, [b, c, [d]], [e, f]]`, o resultado final seria `[a, [b, c, [g, d]], [e, f]]`
5. Implemente uma função que recebe duas listas como entrada e retorna `1` se as listas são iguais e `0` caso contrário.
6. Implemente uma função que retorna a profundidade de uma lista generalizada.
7. Implemente uma função que inverta uma lista generalizada e todas as suas sublistas. Por exemplo, `[a, [b, c]]` se tornaria `[[c, b], a]`.