



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

ESTRUTURA DE DADOS LISTA DE EXERCÍCIOS

1. Escreva um programa em C para implementar uma estrutura de dados do tipo Lista Encadeada de números inteiros (nós que contém um número inteiro). Os seguintes requisitos devem ser contemplados:
 - a. Implemente um TAD ListaEncadeada.
 - b. Para este TAD, forneça as funções:
 - i. create: cria uma ListaEncadeada;
 - ii. add: adiciona um elemento no fim da ListaEncadeada. Esta operação retorna verdadeiro se foi possível adicionar o elemento. Seja *proximo* o ponteiro do último nó da lista e *fim* o ponteiro da lista que aponta o último elemento da lista, tanto *proximo* quanto *ultimo* devem ser atualizados;
 - iii. remove: remover o elemento da ListaEncadeada. Retorna verdadeiro se aquele elemento realmente estava na lista e se foi possível removê-lo. No caso da remoção ter sido efetivada, os ponteiros dos nós ligados ao elemento removido devem ser atualizados;
 - iv. size: retorna a quantidade de elementos inseridos na ListaEncadeada;
 - v. find: recebe um inteiro e retorna a posição deste elemento na ListaEncadeada. Retorna -1 se o elemento não foi encontrado;
 - vi. isEmpty: retorna verdadeiro se a ListaEncadeada está vazia e falso para o caso oposto;
 - vii. clear: remove todos os elementos da ListaEncadeada.
 - c. Descreva a complexidade de cada uma das funções que você implementou.
2. Escreva um programa em C para implementar uma estrutura de dados do tipo Lista Duplamente Encadeada Circular de números inteiros (nós que contém um número inteiro). Os seguintes requisitos devem ser contemplados:
 - a. Implemente um TAD ListaCircular.
 - b. Para este TAD, forneça as funções:
 - i. create: cria uma ListaCircular;
 - ii. add: adiciona um elemento no fim da ListaCircular. Esta operação retorna verdadeiro se foi possível adicionar o elemento. Seja *proximo* o ponteiro do último nó da lista e *fim* o ponteiro da lista que aponta o último elemento da lista, tanto *proximo* quanto *ultimo* devem ser atualizados;
 - iii. remove: remover o elemento da ListaCircular. Retorna verdadeiro se aquele elemento realmente estava na lista e se foi possível removê-lo. No caso da remoção ter sido efetivada, os ponteiros dos nós ligados ao elemento removido devem ser atualizados;

- iv. size: retorna a quantidade de elementos inseridos na ListaCircular;
- v. find: recebe um inteiro e retorna a posição deste elemento na ListaCircular. Retorna -1 se o elemento não foi encontrado;
- vi. isEmpty: retorna verdadeiro se a ListaCircular está vazia e falso para o caso oposto;
- vii. clear: remove todos os elementos da ListaCircular.

c. Descreva a complexidade de cada uma das funções que você implementou.

3. Escreva um programa em C para implementar uma estrutura de dados do tipo Lista Estática Ordenada de números inteiros. Os seguintes requisitos devem ser contemplados:

a. Implemente um TAD ListaOrdenada.

b. Para este TAD, forneça as funções:

- i. create: recebe n e cria uma ListaOrdenada cuja dimensão do vetor de dados é n ;
- ii. add: adiciona um elemento na ListaOrdenada tal que o seu vetor de dados permaneça ordenada com classificação crescente. Esta operação retorna verdadeiro se foi possível adicionar o elemento e falso caso contrário;
- iii. remove: remover o elemento da ListaOrdenada tal que o vetor de dados permaneça ordenado após esta operação. Retorna verdadeiro se aquele elemento realmente estava na lista e se foi possível removê-lo e falso caso contrário;
- iv. size: retorna a quantidade de elementos inseridos na ListaOrdenada;
- v. linearSearch: recebe um inteiro e retorna a posição deste elemento na ListaOrdenada. Retorna -1 se o elemento não foi encontrado. Essa função deve seguir o escopo de implementação de busca sequencial;
- vi. binarySearch: recebe um inteiro e retorna a posição deste elemento na ListaOrdenada. Retorna -1 se o elemento não foi encontrado. Essa função deve seguir o escopo de implementação de busca binária;
- vii. isFull: retorna verdadeiro se a ListaOrdenada está cheia e falso para o caso oposto;
- viii. isEmpty: retorna verdadeiro se a ListaOrdenada está vazia e falso para o caso oposto;
- ix. clear: remove todos os elementos da ListaOrdenada.

c. Descreva a complexidade de cada uma das funções que você implementou.