

3ª Lista de Exercícios

Estrutura de Dados

Prof. Hamilton José Brumatto

Listas Encadeadas e implementações

1. Considere o jogo de eliminação criado na lista de exercícios anterior. Implemente este jogo usando listas encadeadas circulares: Faça com que as entradas de amigos seja via console, criando um item e inserindo-o na lista para cada amigo, e dado o número, remova os eliminados da lista até sobrar somente um jogador, o ganhador. Implemente Inserir e Remover itens como funções.
2. Considere o algoritmo de ordenação por inserção:
Começando pela penúltima posição até a primeira, nós retiramos o elemento desta posição (elemento chave k) e comparamos ele com os próximos elementos na sequência (já ordenados). Se o elemento não for maior que o próximo, ele é inserido justamente antes do próximo, ou seja, ele é maior que todos os anteriores e não é maior que o próximo, logo ele está em sua posição correta.

ORDENACAO_INSERTAO

Entrada: Uma sequência de valores não necessariamente ordenada.

Saída: A mesma sequência em uma ordem crescente

Algoritmo ORDENACAO_INSERTAO(A)

$a \leftarrow last(A)$

repita

$a \leftarrow ant(a)$

$k \leftarrow a$

$rem(k)$

$p \leftarrow a$

enquanto $k > succ(p)$ **faça**

$p \leftarrow succ(p)$

$ins(a, p)$

até que $a = first(A)$

Considere as seguintes informações:

- $first(A)$: é uma função que retorna o primeiro elemento da lista A .
- $last(A)$: é uma função que retorna o último elemento da lista A .
- $ant(a)$: é uma função que retorna da lista o elemento que antecede o elemento a .
- $succ(a)$: é uma função que retorna da lista o elemento que sucede o elemento a .
- $rem(a)$: é uma função que remove da lista o elemento a (sem excluí-lo da memória).
- $ins(a, p)$: é uma função que insere na lista o elemento a antes do elemento p (p pode ser NULL).
- a comparação $a > b$ é uma comparação entre os conteúdos dos elementos a e b .

Pede-se:

- (a) Armazene um conjunto de elementos na forma de uma lista encadeada, aplique o algoritmo descrito e imprima o resultado. Considere os elementos como números racionais.
 - (b) Considere que andar na lista ($p = p \rightarrow prox$) tem o custo de 1 (em valores de operações), e somente esta operação tem custo. Qual seria o custo deste algoritmo de ordenação em função do tamanho n da lista, considerando o pior caso, ou seja, o *Enquanto* é realizado até o último elemento da lista.
3. **DEQUE** (Double Ended QUEue) fila com dois extremos: DEQUE é uma fila. Nós vamos ver mais à frente o conceito de fila. Mas vamos construir um DEQUE simples, apenas duas operações: Inserir e Remover. Inserir insere um elemento no extremo da fila (lista), e remover remove o elemento do extremos da lista (fila). A operação inserir deve receber duas informações: qual o valor e qual o extremo. Também a operação remover. Fica claro que inserir e remover em um deque pode ser implementado facilmente usando listas duplamente encadeadas. Construa um DEQUE com estas duas operações.