

2º Aulão de ED

Pilha, Lista Contígua e Lista Encadeada

Samuel Abreu
Filipe Brinati

Pilha

1. (25 Pontos) Considere o TAD **PilhaEncad** para representar uma **pilha encadeada** (detalhes deste TAD no enunciado). Implemente a função `bool balanceado(char vet[], int n)` que utiliza uma pilha de caracteres para testar se o vetor **vet**, contendo apenas {, }, [, e] é balanceado (retorna **true**) ou não (retorna **false**). Usar uma pilha com a seguinte estratégia: enquanto percorre os caracteres do vetor:
 - se o caractere atual for { ou [, então empilhar o caractere;
 - se o caractere atual for } ou] e o topo for o seu par correspondente ({ ou [, respectivamente), então desempilhar;
 - qualquer outro caso significa que o vetor não está balanceado.

O vetor está balanceado se, após a sua completa varredura, a pilha ficar vazia.

Lista Contígua

2. (25 Pontos) Considere o TAD **ListaCont** para representar uma **lista contígua** de valores inteiros (detalhes de implementação no enunciado). Implemente a função `bool ListaCont::insereIntervalo(int a, int b)` que insere o intervalo [a, b] no final da lista, **sem utilizar operações pré-definidas** (exceto as disponíveis, aumente a capacidade máxima do vetor se necessário). Retorne **true** se a capacidade máxima foi aumentada, sem que nenhum dado atualmente representado na lista seja perdido. Ao final retornar se a capacidade máxima foi ou não aumentada.

Lista Dup. Encadeada

3. (25 Pontos) Considere o TAD **ListaDupEncad** para representar uma **lista duplamente encadeada** de valores inteiros (detalhes de implementação no enunciado). Implemente a função `void ListaDupEncad::removeMaior()` que remove o nó com maior valor da lista. Retorne **true** se a lista não estava vazia, **false** caso contrário. Percorrer a lista uma única vez. Imprima o resultado no final.

Lista Encadeada

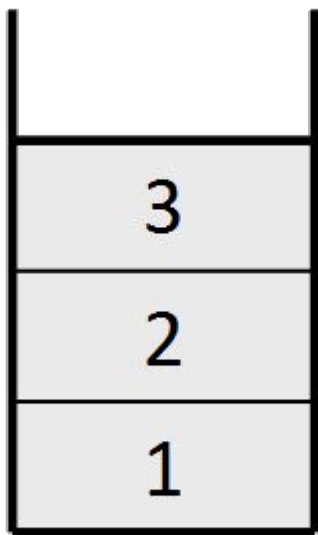
4. (25 Pontos) Considere o TAD **ListaEncad** que representa uma **lista simplesmente encadeada** de valores inteiros (detalhes de implementação no enunciado). Implemente a função `void ListaEncad::removeBloco(int a, int b)` que remove o bloco da lista com valor a e o elemento com valor b, inclusive. Retorne **true** se a lista não estava vazia, **false** caso contrário. Assuma que: (i) o elemento com valor a não está na lista; (ii) havendo a na lista então nenhum elemento deve ser removido; e (iii) havendo um elemento com o valor a mas não havendo um com b então todos os elementos a partir de a devem ser removidos.

1. Pilha

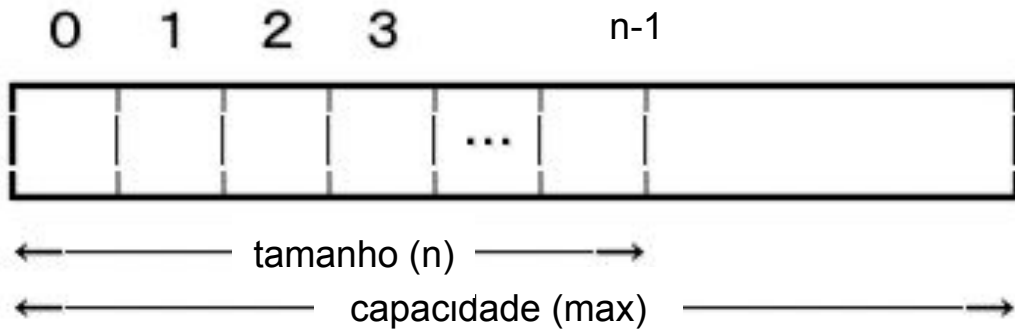
Empilhar

Desempilhar

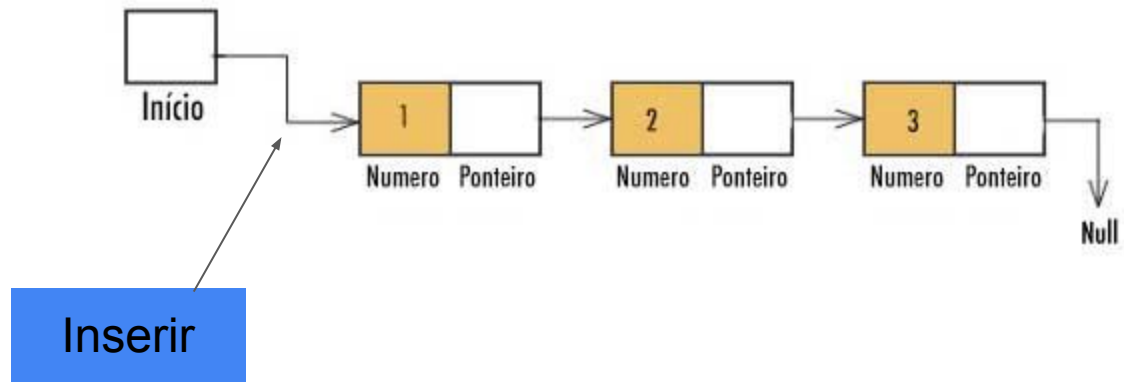
Topo

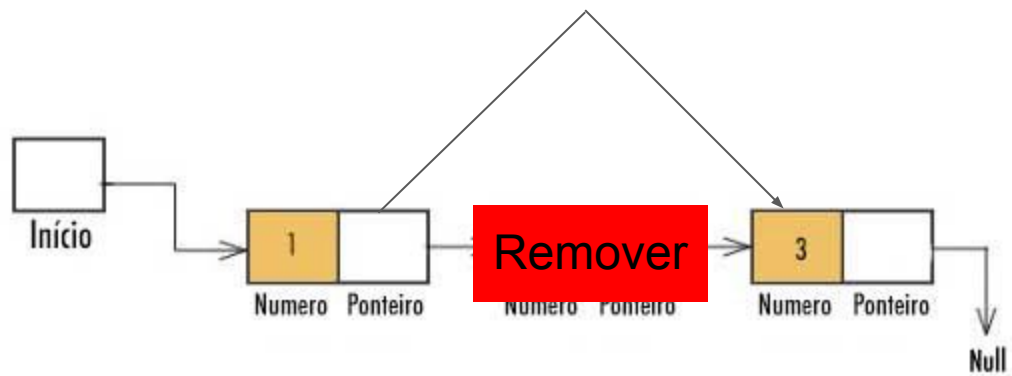


2. Lista Contígua



3. Lista Encadeada





4. Lista Dupla Encadeada

Inserir

