



Exercícios de Estrutura de Dados – Vetor, Lista e Sequencia

1. Implemente uma classe VetorArray, que implemente a interface Vetor (baixe da página da disciplina). Faça uma tabela com os tempos de execução de cada método. Use a implementação da classe java.util.Vector de JAVA.

2. Considere o seguinte algoritmo para o método insertAfter(p,e):

```
public Nó insertAfter(n, e){
    DNode no1 = n;
    DNode no2 = new DNode();
    no2.setElement(e);
    no2.setPrev(no1);
    no2.setNext(no1.getNext());
    no1.setNext(no2);
    (no1.getNext()).setPrev(no2);
    return no2;
}
```

O método está certo? Justifique.

Dica: Desenhe as referências após cada linha do algoritmo

3. Desenvolva um método recursivo public Position find(int i) que retorne a Nó de um inteiro i em uma lista de inteiros. O método deverá disparar a exceção ValueNotFoundException.

4. Desenhe uma figura que representa os elementos de uma sequência S (inicialmente vazia) depois das seguintes operações:

- (a) S.insertLast(a)
- (b) S.insertFirst(b)
- (c) S.insertAtRank(1,c)
- (d) S.remove(S.after(S.first()))
- (e) S.insertAfter(S.last(),d)

5. Considere uma sequência implementada com uma lista duplamente encadeada (os nós são da classe NoSeq) e que possui nós sentinelas que indicam os primeiro e o último elementos. Preencha a linha em branco de uma implementação do método insertBefore em JAVA. Assuma que os nós implementam a interface NóSeq.

```
public NóSeq insertBefore(NóSeq p, Object e){
    NoSeq temp = new NoSeq();
    temp.setElement(e);
    temp.setNext(p);
    temp.setPrev(p.getPrev());

    _____
    p.setPrev(temp);
    return temp;
}
```

6. Implemente uma sequência usando o objeto Vector do JAVA.
7. Implemente uma classe VetorLista, que implemente a interface Vetor, e que use como representação interna uma lista duplamente ligada. Faça tabela comparando os tempos de execução das duas implementações do Vetor (array e Lista Encadeada).
8. Faça uma tabela com os tempos de execução dos métodos de Sequência implementados com arranjo e com Lista duplamente ligada. Descreva os algoritmos dos métodos que possuem tempos de execução diferente nas duas implementações e explique porque os tempos são diferentes.
9. Um programa P usa uma implementação de sequência como seu componente principal. Sabe-se que P usa as operações atRank, insertAtRank e remove em ordem não determinada. Sabe-se também que P realiza n_2 operações atRank, $2n$ operações insertAtRank e n operações remove. Qual implementação do TAD sequência deve ser usada para obter maior eficiência: a implementação baseada em array ou a implementação baseada em lista duplamente encadeada? Justifique sua resposta.
10. Considere um novo método no TAD Sequencia (implementado com lista duplamente ligada), chamado makeFirst(p), que move o elemento na posição p de uma sequência S para a primeira posição de S. O método deve manter a ordem relativa de todos os outros elementos e deve executar em tempo $O(1)$.
11. Adicione o método shrinkToFit() à sua classe VetorArray, que substitua o array atual por um array exatamente do tamanho do Vetor.
12. Faça um programa em JAVA para testar as classes que implementem a interface List. Imagine uma lista de itens, onde você pode acrescentar e remover itens no fim ou no início da lista. Além disso, você tem um "cursor" que indica o item em evidência, e você pode inserir itens antes ou depois deste item, além de poder removê-lo. Também considere botões de navegação, nos quais você pode mudar o item em evidência, também pode ir para o primeiro ou o último item.
13. Problema da reordenação: Dada uma sequência de n elementos, podemos querer que estes elementos mantenham uma relação de ordem. Consideramos que quaisquer dois elementos consecutivos possam ser comparados (use um comparador) para determinar se estão na ordem desejadas e, se não estiverem, eles podem ser trocados de posição. Dica veja o algoritmo da Bolha.
Responda:
 - (a) Qual a ordem de complexidade do algoritmo e porque?
 - i. Se o TAD sequência for implementado usando um array.
 - ii. Se o TAD sequência for implementado usando uma lista duplamente encadeada.
 - (b) Implemente o algoritmo.

Sites sobre método da bolha:

- <http://math.hws.edu/TMCM/java/xSortLab/>
- <http://www.cs.rit.edu/~atk/Java/Sorting/sorting.html>

Dica para a prática: faça um programa que leia números inteiros de um arquivo e escreva um outro arquivo com os números ordenados. Os nomes dos arquivos podem ser parâmetros passados para o método main.