M3L: Pesquisa e ordenação

Murilo Dantas

PROBLEMAS DE REVISÃO

1. **Uma pesquisa binária supõe que os dados estejam:**
   a. **Organizados sem uma ordem específica.**
   b. **Ordenados.**

2. **Uma ordenação por seleção faz no máximo:**
   a. **n^2 trocas de itens de dados.**
   b. **n trocas de itens de dados.**

3. **O comportamento no melhor caso da ordenação por inserção e a ordenação por bolha modificada é:**
   a. **Linear.**
   b. **Quadrático.**
   c. **Exponencial.**

PROBLEMAS DE AVALIAÇÃO

1. **Implemente um programa estruturado e recursivo para pesquisa linear. Faça uma função de busca chamada** pesquisaLR **que receba como parâmetro o valor a ser encontrado e a referência do vetor onde a busca será efetuada. A função retornará -1, caso não encontre o item, ou retornará o índice, caso o encontre.**

2. **Implemente um programa estruturado e recursivo para pesquisa binária. Faça uma função de busca chamada** pesquisaBR **que receba como parâmetro o valor a ser encontrado e a referência do vetor onde a busca será efetuada. A função retornará -1, caso não encontre o item, ou retornará o índice, caso o encontre.**

3. **Um vetor de tamanho n, pode conter elementos do alfabeto e numerais 0 a 9. Escreva um algoritmo que seja capaz de localizar, pelo método binário, um caractere fornecido pelo usuário. Se esse caractere for uma letra, o usuário poderá fornecê-la para a busca no formato maiúsculo ou minúsculo.**

4. **Faça um programa que cadastre n produtos. Para cada produto devem ser cadastrados os seguintes dados: código, descrição e preço. Use um método de ordenação e em seguida calcule e mostre quantas comparações devem ser feitas para encontrar um funcionário pelo código:**
   a. **Usando busca sequencial.**
   b. **Usando busca binária.**

5. **Faça um programa que cadastre n números, não permitindo números repetidos. Ordene-os e, em seguida, verifique se o número digitado pelo usuário está no vetor. Caso encontre, verifique se está numa posição par ou ímpar do vetor:**
    a. **Usando busca sequencial.**
    b. **Usando busca binária.**

6. **Dada uma tabela de horários de ônibus que fazem viagens para as diversas cidades do Estado, escreva um programa que possibilite a localização dos horários de saída e de chegada quando se forneça o destino.**

7. **Implemente uma versão generalizada para busca binária numa matriz $m \times n$.**

8. **Elabore uma matriz com 500 linhas e 50 colunas, que deverá ser preenchida com números inteiros aleatórios na faixa de 1 a 10.000. Faça a busca, pelo método binário, de um elemento sorteado, indique a quantidade de elementos iguais a este presente na matriz e indique a posição (ou as posições, caso aja repetição) em que ele se encontra (i, j).**

9. **Faça um programa que cadastre o nome e o salário de n funcionários. Usando um método de ordenação diferente para cada item a seguir, liste todos os dados dos funcionários das seguintes formas:**
    a. **Em ordem crescente de salário;**
    b. **Em ordem decrescente de salário;**
    c. **Em ordem alfabética.**

10. **Faça um programa que cadastre n números, ordene-os pelo *bubbleSort* e em seguida encontre e mostre:**
    a. **O menor número e quantas vezes ele aparece no vetor.**
    b. **O maior número e quantas vezes ele aparece no vetor.**

11. **Faça um programa que cadastre n alunos. Para cada aluno devem ser cadastrados: nome, nota1 e nota2. Primeiro, liste todos os alunos cadastrados, ordenando-os pela média ponderada das notas, tendo a primeira nota peso 2 e a segunda, peso 3. Em seguida, ordene os alunos, de forma crescente, pela nota1, e liste-os. Finalmente, considerando que, para ser aprovado, o aluno deve ter no mínimo média 7, liste, em ordem alfabética, os alunos reprovados. Em cada ordenação use um algoritmo diferente.**

12. **Desenvolva uma aplicação que, dados dois vetores de inteiros quaisquer com tamanho de 20 elementos, gere um terceiro com os elementos de ambos em ordem crescente, usando o *mergeSort*. Apresente o resultado final.**

13. **Crie uma aplicação que implemente uma matriz quadrada com n números inteiros, os quais devem ser fornecidos aleatoriamente pelo usuário. Implemente um menu com duas opções:**
    a. **Colocar os elementos em ordem crescente (use o *insertionSort*).**
    b. **Colocar os elementos em ordem decrescente (use o *selectionSort*).**

14. **Elabore um programa que armazene os seguintes dados de n pessoas: nome, idade e sexo. O programa deve apresentar os dados em:**
    a. **Ordem crescente alfabética de nome (use o *quickSort*).**
    b. **Ordem decrescente de idade (use o *bubbleSort*).**

15. **Crie um vetor que armazene dados de n funcionários de uma empresa. Deverão ser considerados os dados: código funcional, nome, salário e data de admissão. Elabore um programa que: preencha o vetor com os dados fornecidos pelo usuário e ordene de forma crescente os elementos pelo campo de código funcional, usando o *quickSort*.**

16. **Crie um vetor que armazene dados de n casas numa imobiliária. Deverão ser considerados os dados: código, bairro, tamanho em m^2, valor de venda e valor de aluguel. Elabore um programa que: preencha o vetor com os dados fornecidos pelo usuário e ordene de forma decrescente os elementos pelo campo de venda, usando o *bubbleSort*.**

17. **Crie uma aplicação que permita inserir cerca de 10 mil números inteiros aleatórios de 1 a 10 mil num vetor de inteiros. Registre o tempo de início e término da operação de ordenação e compare essas diferenças entre os algoritmos *bubbleSort*, *insertionSort* e *quickSort*. Comente as diferenças e considere testar com números diferentes de elementos. Dica: quando tiver rodando os algoritmos, evite executar outros programas na máquina.**

18. **Crie uma aplicação que permita inserir cerca de 8 mil números inteiros aleatórios de 1 a 8 mil num vetor de inteiros. Faça um comparativo considerando o número de trocas realizadas entre os algoritmos *selectionSort*, *mergeSort* e *quickSort*. Comente as diferenças e considere testar com números diferentes de elementos. Dica: quando tiver rodando os algoritmos, evite executar outros programas na máquina.**