



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

BACHARELADO EM SISTEMAS DE INFORMAÇÃO
ESTRUTURA DE DADOS II
PROF. HÉLDER SEIXAS LIMA

Lista de Exercícios: algoritmos de ordenação

1) Implemente os algoritmos que se pede em alguma linguagem de programação. Também encontre a função de complexidade e comportamento assintótico da sua implementação.

- a) Shellsort
- b) Quicksort
- c) Heapsort

2) Dado que existe necessidade de ordenar arquivos de tamanhos diversos, podendo também variar o tamanho dos registros de um arquivo para outro, apresente uma discussão sobre quais algoritmos de ordenação você escolheria diante das diversas situações colocadas acima.

Que observações adicionais você apresentaria caso haja:

- a) restrições de estabilidade ou
- b) de intolerância para o pior caso (isto é, a aplicação exige um algoritmo eficiente mas não permite que o mesmo eventualmente leve muito tempo para executar).

3) Invente um vetor-exemplo de entrada para demonstrar que Ordenação por Seleção é um método instável. Mostre os passos da execução do algoritmo até que a estabilidade é violada. Observe que quanto menor for o vetor que você inventar, mais rápido você vai resolver a questão.

4) O Quicksort não é um algoritmo estável. Que tipo de transformação você poderia fazer para que ele se transforme em um algoritmo estável? Qual o custo é acrescido à complexidade do algoritmo com essa transformação?

5) Qual algoritmo de ordenação você usaria para cada um dos seguintes casos abaixo. Explique suas respostas.

- a) A ordenação original de elementos com chave idêntica precisa ser mantida.
- b) O tempo de execução não deve apresentar grandes variações para nenhum caso.
- c) A lista a ser ordenada já está bem próxima da ordem final.
- d) Os elementos a serem ordenados são muito grandes.

6) Ordene as letras da palavra UMDOIS mostrando o conteúdo do vetor a cada passo intermediária. Utilize os seguintes algoritmos de ordenação:

- a) Seleção. Liste o vetor para cada elemento que atinja sua posição definitiva até o momento.
- b) Inserção. Liste o vetor para cada elemento incluído na ordenação parcial até o momento.
- c) Shellsort. Use 1, 3, 5, 13 como sequência de valores para h . Liste o vetor para cada novo valor de h .
- d) Quicksort, usando o elemento à esquerda da partição como pivô. Liste o vetor para cada nova partição.
- e) Quicksort, usando o elemento $(\text{esquerda} + \text{direita})/2$ como pivô. Liste o vetor para cada nova partição.
- f) Heapsort. Liste o vetor após a construção do *heap* e após a refazê-lo.

7) Mostre as etapas de ordenação das palavras abaixo por meio do Quicksort. Considere que o pivô de cada partição corresponde à $(\text{esquerda} + \text{direita})/2$.

- a) ABABABA
- c) QUICKSORT

8) O objetivo deste exercício é fazer um estudo comparativo de diversas implementações do algoritmo Quicksort. Para tanto, você deverá implementar as seguintes versões do algoritmo:

- a) Quicksort recursivo;
- b) Quicksort recursivo com interrupção da partição para a ordenação de sub-vetores menores que M elementos utilizando ordenação por inserção. Determine empiricamente o melhor valor de M para um arquivo gerado aleatoriamente com 1000 (mil) elementos;
- c) Melhore a versão 2 utilizando a técnica de mediana-de-três elementos para escolha do pivô.
- d) Quicksort recursivo com verificação a cada chamada recursiva para verificar se o vetor já está ordenado.

Qual a função de complexidade de cada um desses algoritmos?

Gere massas de testes para testar e comparar cada uma das implementações. Determine experimentalmente o número de (i) comparações, (ii) movimento de elementos e (iii) tempo real de execução para cada um dos cinco métodos de ordenação indicados acima. Use sua criatividade para criar arquivos de teste interessantes. Faça tabelas e/ou gráficos para mostrar e explicar os resultados obtidos.

9) O objetivo deste problema é projetar uma estrutura de dados para um conjunto S . A estrutura de dados projetada deve implementar eficientemente as seguintes operações:

a) $\text{Insere}(b, S)$: Insere o elemento b em S (isto vai adicionar uma nova cópia de b em S se já existia alguma).

b) $\text{RetiraMin}(x, S)$: retira de S o menor elemento (pode haver mais de um), retornando seu valor em x .

Descreva uma estrutura de dados e como implementar as operações Insere e RetiraMin de modo que estas operações sejam executadas, no pior caso, em $O(\log n)$.

10) Faça um programa que leia n nomes inserindo-os em uma lista de forma ordenada utilizando a ideia do algoritmo da inserção. No final, o programa deve mostrar todos os nomes ordenados alfabeticamente.

11) Crie um programa que dado uma string, coloque as letras dela em ordem crescente pelo algoritmo da seleção.

12) Dada a sequência de números: 3 4 9 2 5 8 2 1 7 4 6 2 9 8 5 1, ordene-a em ordem crescente segundo os seguintes algoritmos, apresentando a sequência obtida após cada passo do algoritmo:

a) Shellsort

b) Quicksort

c) Heapsort

13) João diz ter desenvolvido um algoritmo que é capaz de ordenar qualquer conjunto de n números reais, fazendo apenas $O(n^{3/2})$ comparações. Você compraria este algoritmo? Justifique.

14) Um amigo lhe diz que é capaz de ordenar qualquer conjunto de 8 números com no máximo 16 comparações. O seu amigo está falando a verdade ou mentindo? Justifique.

15) Desenvolva um algoritmo usando um *heap* para encontrar os maiores k números num grande vetor não classificado de n números (onde $n > k$).