Algoritmos e Estruturas de Dados II
Lista de Exercícios de Ordenação

Bacharelado em Análise de Sistemas, DCT–UFMS, 6/4/2005

1. Ilustre a operação de todos os algoritmos de ordenação que estudamos sobre o vetor $A = \langle 3, 41, 52, 26, 38, 57, 9, 49 \rangle$.

2. A ordenação por inserção pode ser expressa como um procedimento recursivo da seguinte forma. Para ordenar o vetor $A[1..n]$, ordenamos recursivamente o vetor $A[1..n-1]$ e então inserimos $A[n]$ no vetor ordenado $A[1..n-1]$. Escreva um algoritmo de ordenação por inserção recursivo que utiliza essa idéia.

3. Escreva um algoritmo para o $\textsc{Intercala}(A, p, q, r)$.

4. Suponha que estamos comparando implementações da ordenação por inserção e da ordenação por intercalação em um mesmo computador. Para entradas de tamanho n, a ordenação por inserção realiza $8n^2$ passos enquanto que a ordenação por intercalação realiza $64n \log n$ passos. Para quais valores de n a ordenação por inserção é melhor que a ordenação por intercalação?

5. Seja $A[1..n]$ um vetor de n números distintos. Se $i < j$ e $A[i] > A[j]$ então o par (i, j) é chamado uma **inversão** de A.

    (a) Liste as cinco inversões de $A = \langle 2, 3, 8, 6, 1 \rangle$.

    (b) Qual vetor com elementos do conjunto $\{1, 2, \ldots, n\}$ tem o maior número de inversões? Quantas são?

    (c) Forneça um algoritmo eficiente que determina o número de inversões em qualquer permutação de n elementos.
    Sugestão: modifique o algoritmo de ordenação por intercalação.

6. Qual valor de q o algoritmo $\textsc{Partição}$ devolve quando todos os elementos no vetor $A[p..r]$ têm o mesmo valor?

7. Como você modificaria o algoritmo de ordenação por particionamento para ordenar elementos em ordem não crescente?

8. Bancos freqüentemente registram transações sobre uma conta corrente na ordem dos tempos em que essas transações ocorrem, mas muitas pessoas preferem receber seus extratos bancários com seus cheques listados na ordem dos números dos cheques emitidos. As pessoas usualmente emitem cheques em ordem numérica e os comerciantes descontam

esses cheques em tempo razoável. O problema de converter uma lista de cheques ordenada por tempo de transação para uma lista de cheques ordenada por número de cheques é portanto o problema de ordenar uma entrada quase ordenada. Argumente que, neste caso, o algoritmo de ordenação por inserção tende a ser melhor que os algoritmos de ordenação por intercalação ou por particionamento.

9. Considere a seguinte variação do algoritmo PARTIÇÃO, devido a N. Lomuto. Para particionar um vetor $A[p..r]$, esta versão aumenta duas regiões $A[p..i]$ e $A[i+1..j]$, tal que todo elemento na primeira região e menor ou igual a $x = A[r]$ e todo elemento na segunda região é maior que x.

PARTIÇÃO–LOMUTO(A, p, r)

$x \leftarrow A[r]$
$i \leftarrow p - 1$
para $j \leftarrow p$ **até** r **faça**
    se $A[j] \leqslant x$ **então**
        $i \leftarrow i + 1$
        TROCA$(A[i], A[j])$
se $i < j$ **então**
    devolva i
senão
    devolva $i - 1$

Execute o algoritmo PARTIÇÃO–LOMUTO sobre o vetor A do exercício 1. Qual(is) a(s) diferença(s) entre o algoritmo PARTIÇÃO e o algoritmo PARTIÇÃO–LOMUTO? Verifique, por exemplo, qual o número máximo de vezes que um elemento pode ser movido por esses algoritmos. Argumente que o algoritmo PARTIÇÃO–LOMUTO tem tempo de execução $O(n)$ sobre um vetor de n elementos.

10. Três professores juntaram-se e propuseram o seguinte algoritmo de ordenação:

ORD–RÁPIDO(A, i, j)

se $A[i] > A[j]$ **então**
    TROCA$(A[i], A[j])$
se $i + 1 < j$ **então**
    $k \leftarrow (j - i + 1)$ **div** 3
    ORD–RÁPIDO$(A, i, j - k)$
    ORD–RÁPIDO$(A, i + k, j)$
    ORD–RÁPIDO$(A, i, j - k)$

Execute o algoritmo ORD–RÁPIDO sobre o vetor A do exercício 1. Qual o tempo de execução do algoritmo ORD–RÁPIDO? Compare esse tempo com os algoritmos de ordenação visto em sala.