UNIVERSIDADE DE SÃO PAULO – ICMC
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO

SCC601 – Introdução a Ciência da Computação II
Rosane Minghim

Estagiária P.A.E: Nathalie Portugal **Monitor**: Athila Quaresma Santos

Lista de Exercícios Nº 4
Ordenação

1. Um vetor v[p..r] está "arrumado" se existe j ∈ [p, r] tal que v[p .. j-1] ≤ v[j] < v[j+1 .. r]. Escreva um algoritmo que decida se v[p..r] está arrumado. Em caso afirmativo, o seu algoritmo deve devolver o valor de j.

2. Um programador inexperiente afirma que a seguinte implementação da função de partição rearranja o vetor v[p..r], com p < r, e devolve um índice j ∈ [p, r -1] tal que v[p..j] ≤ v[j+1 ..r].

```
int part (int v[], int p, int r)
{
    int q, i, j, t;
    i = p; q = (p + r) / 2; j = r;
    do
    {
        while (v[i] < v[q]) i++;
        while (v[j] > v[q]) j-;
        if (i <= j)
        {
            t = v[i], v[i] = v[j], v[j] = t;
            i++, j-;
        }
    } while (i <= j);
    return i;
}
```

Mostre um exemplo onde essa função não dá o resultado esperado. E se trocarmos return i por return i-1? É possível fazer algumas poucas correções de modo que a função dê o resultado esperado?

3. Considere a ordenação de n números armazenados no arranjo A, localizando primeiro o menor elemento de A e permutando esse elemento contido em A[1]. Em seguida, encontre o segundo menor elemento de A e o troque pelo elemento A[2]. Continue dessa maneira para os primeiros n - 1 elementos de A.

Tal algoritmo é conhecido como ordenação por seleção. Escreva o código para esse algoritmo. Por que ele só precisa ser executado para os primeiros n – 1 elementos, e não para todos os elementos? Forneça os tempos de execução do melhor caso e do pior caso da ordenação por seleção em notação O.

4. Ilustre a operação SELECTION-SORT sobre o arranjo A = (89,1,35,78,4,22,100).

SCC601 – Introdução a Ciência da Computação II
Rosane Minghim

Estagiária P.A.E: Nathalie Portugal **Monitor**: Athila Quaresma Santos

5. Ilustre a operação de INSERTION-SORT no arranjoA = (31,41, 59, 26, 41, 58).

6. Reescreva o procedimento INSERTION-SORT para ordenar em ordem não crescente, em vez da ordem não decrescente.

7. Ilustre a operação de ordenação por intercalação sobre o arranjo A = (3, 41, 52, 26, 38, 57, 9, 49).

8. Quais são os números mínimo e máximo de elementos em um heap de altura h?

9. Ilustre a operação de HEAPSORT sobre o arranjo B = (5, 13, 2, 25, 7, 17, 20, 8, 4).

10. Escreva uma rotina combine(x) que aceite um vetor x no qual as sub-árvores enraizadas em x[1] e x[2] sejam heaps e que modifique o vetor x de maneira que ele represente um único heap.

11. Ilustre a operação de PARTIÇÃO referente ao algoritmo QUICKSORT sobre o arranjo C = (13,19,9, 5, 12,8, 7,4, 11, 2,6, 21).

12. Discuta como a escolha do pivô pode influenciar no desempenho do método quicksort. Proponha estratégias para a escolha do pivô, visando melhorar seu desempenho.

13. De que maneira você modificaria QUICKSORT para fazer a ordenação em ordem não crescente.

14. Modifique a rotina quicksort para usar uma classificação por inserção simples quando um sub-vetor tiver um tamanho menor que s.

15. Crie um algoritmo chamado quickfind baseado no quicksort para que, em vez de ordenar uma sequência de números inteiros, ele nos retorne o k-ésimo menor elemento dessa sequência. Por exemplo: Suponha que os elementos S = {7, 1, 3, 10, 17, 2, 21, 9} estejam armazenados nessa ordem em um vetor e que desejamos obter o quinto maior elemento dessa sequência. Então, uma chamada como quickfind(S,0,7,5), deverá retornar o número 9, onde S é o nome do vetor, 0 e 7 são, respectivamente, a menor e a maior posição do vetor e 5 indica que desejamos o quinto menor elemento.

Obs.: Você não deve ordenar a sequência e depois tomar o k-ésimo elemento.

16. Ilustre a operação de COUNTING-SORT sobre o arranjo D = ***(6, 0,*** **2,** O, **1,** 3, 4, 6, **1,** 3, 2).

17. Ilustre a operação de RADIX-SORT sobre a seguinte lista de palavras em inglês: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, **EAR,** TAR, DIG, BIG,TEA, NOW, FOX.

18. Quais dos seguintes algoritmos de ordenação são estáveis: ordenação por inserção, ordenação por intercalação, heapsort e quicksort?

19. Determine qual das seguintes classificações é mais eficiente:

    i. a classificação por inserção simples,
    ii. a classificação por seleção direta,

iii. a classificação por intercalação.

20. Compare o desempenho dos métodos estudados, identificando em que casos você utilizaria cada um deles. Justifique suas afirmações.

21. Vamos supor que estamos comparando diferentes implementações de ordenação por inserção e ordenação por intercalação na mesma máquina. Para entradas de tamanho n, a ordenação por inserção é executada em 8n2 etapas, enquanto a ordenação por intercalação é executada em 64n log n etapas. Para que valores de n a ordenação por inserção supera a ordenação por intercalação?