

MAC 122 – Princípios de Desenvolvimento de Algoritmos**Segundo semestre de 2007**

Lista de Exercícios – Ordenação e Busca de padrões

Entrega: 18 de novembro¹

1. Ordene a seqüência abaixo usando os métodos Mergesort, Quicksort, Heapsort e Bubblesort:

12 23 5 9 0 4 1 12 21 2 5 14

Em cada um dos casos, qual o número de comparações e de trocas feitas durante a ordenação?

2. Faça uma função Merge que recebe vetores ordenados A (com m elementos) e B (com n elementos) e devolve um vetor ordenado C (com $m+n$ elementos) que resulta da intercalação de A e B . Faça duas versões do algoritmo: uma iterativa e uma recursiva. Encontre instâncias (dados) em que o algoritmo acima faz:
 - (a) m comparações;
 - (b) n comparações;
 - (c) $m+n-1$ comparações.
3. Considere o seguinte trecho de programa:

```
for (i = 0; i < n; i++) cont[i] = 0;
for (i=0; i< n; i++)
    for (j = 0; j < n; j++)
        if (V[j] < V[i]) cont[i]++;
```

- (a) Escreva um algoritmo que ordene um vetor V utilizando o trecho acima. Observe o que ocorre se V tiver elementos repetidos.
 - (b) Quantas comparações e quantas movimentações envolvendo os elementos do vetor V são feitas no melhor caso? Quantas no pior caso? Quantas no caso médio?
4. Considere as seguintes funções de ordenação de um vetor A :

¹A lista pode ser entregue no paca até esta data, ou entregue em papel até às 9:00 de segunda-feira, 19/11/2007 na sala 108C, ou na portaria do bloco C

```

void ordena (vetor A, int n)
{
    int i, j, min;

    for(i = 0; i < n - 1; i++){ /* I */
        min = i;
        for (j = i+1; i < n; i++)
            if (A[j] < A[min]) /* II */
                min = j;
        troca (A, i, min); /* III */
    }
}

```

```

void class (vetor A, int n)
{
    int i, j;

    for (i = 1; i < n; i++) /* I */
        for (j = i; j > 1; j--)
            if (A[j] < A[j-1]) /* II */
                troca (A, j, j-1); /* III */
}

```

Para cada um dos algoritmos responda às seguintes perguntas.

- (a) Para cada um dos algoritmos, qual a situação do vetor A a cada passagem pelo ponto I? Justifique.
 - (b) Baseado em sua resposta no item acima, mostre que cada uma das funções de fato ordena o vetor A.
 - (c) Qual o número máximo e mínimo de vezes que a comparação II é executada e em que situações ocorre?
 - (d) Idem para o comando III, que troca elementos.
5. Considere uma matriz $A_{m \times n}$ em que cada uma das linhas e cada uma das colunas estão ordenadas em ordem crescente (veja exemplo abaixo). Escreva um algoritmo que recebe uma matriz com esta propriedade e um elemento x e faz no máximo $m + n$ comparações para verificar se x está ou não na matriz.

Exemplo:

$$\begin{pmatrix} 12 & 20 & 21 & 42 \\ 15 & 22 & 25 & 51 \\ 16 & 28 & 31 & 94 \\ 23 & 32 & 51 & 98 \\ 77 & 91 & 93 & 123 \end{pmatrix}$$

6. Considere o texto $t = 77368910083723459572231252382343494343531$ e o padrão $s = 2345957$. Considere que a função de hash usada no algoritmo de busca de padrões de Karp-Rabin será

$h(x) = x \% 11$, onde x é uma janela de t de comprimento 7 (tamanho de s). Mostre o funcionamento do algoritmo para esta instância.

7. Calcule o vetor `apr` do KMP para o padrão $s = ababbabbababbababbabb$.
8. Considere o vetor `apr` como uma função definida em $\{0, 1, \dots, m-1\} \rightarrow \{-1, 0, 1, \dots, m-1\}$. Formule um limite superior para o número de vezes que a função `apr` pode ser aplicada (até o resultado ser -1). Construa uma instância que mostra que seu limitante superior ocorre.