

## MAC122 – Princípios de Desenvolvimento de Algoritmos EP no. 3

### **Algoritmos de Ordenação:**

O objetivo deste exercício-programa é o de implementar e estudar o comportamento médio dos algoritmos de ordenação. Devem ser considerados:

- 1) Ordenação por Inserção
- 2) Ordenação Bubblesort
- 3) Ordenação por Seleção Simples
- 4) Heapsort
- 5) Quicksort
- 6) Mergesort

Você deve construir um programa (em C) para testar a eficiência média dos algoritmos acima. Queremos saber quantas comparações, quantas trocas e quanto tempo cada um dos 6 algoritmos acima precisa, em média, para ordenar vários vetores de tamanho  $n$ , escolhidos aleatoriamente. Para isso seu programa deve:

- 1 - Inicialmente preencher um vetor de números inteiros aleatoriamente,
- 2 - Ordenar o mesmo vetor uma vez com cada um dos 6 algoritmos acima.
- 3 - Contar, para cada algoritmo, o número de comparações e trocas realizadas,
- 4 - Cronometrar o tempo que cada algoritmo gasta.

O ideal é rodar várias vezes para cada valor de  $n$  os algoritmos, de modo a considerar uma estatística de execução, principalmente para valores de  $n$  pequenos. Este processo deve ser repetido para valores de  $n$  crescentes. Utilize seu bom senso para determinar quantas vezes repetir cada teste e até que valor de  $n$  chegar. Uma regra razoável é não utilizar mais do que 5 minutos de tempo de execução no total para todos os testes efetuados. Observe empiricamente como os resultados teóricos se verificam na prática.

Entregue, além do programa, um relatório do seu programa que contém as seguintes informações:

- 1) O número de testes realizados para cada valor de  $n$ .
- 2) O tempo médio calculado para cada algoritmo para cada valor de  $n$ .
- 3) O número médio de comparações realizadas para cada algoritmo para cada valor de  $n$ .
- 4) O número médio de trocas realizadas para cada algoritmo para cada valor de  $n$ .
- 5) Os mesmos valores (2,3 e 4) divididos por  $n \cdot \log(n)$  e por  $n^2$ .

Comente também as suas conclusões a respeito de:

- 1) A ordem do tempo de processamento de cada algoritmo no caso médio (para valores relativamente grandes de  $n$ ).
- 2) A relação entre os algoritmos para valores diferentes de  $n$ .
- 3) A relação entre o número de comparações, número de trocas e o tempo gasto por cada algoritmo. Ou seja, a contagem destas operações básicas é uma boa maneira de se classificar o tempo do algoritmo?

Para realizar medições de tempo, use as funções da biblioteca padrão `time.h`:

```
#include <time.h>

clock_t start, end;
double cpu_time_used;

start = clock();
... /* Código a ser cronometrado */
end = clock();
cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;
```

***Data limite para Entrega: 24/11/2011.***