

Trabalho 1

Como primeiro trabalho, você deverá escrever um programa em C para analisar o tempo de execução de algoritmos de ordenação. O programa deverá fazer a ordenação de um conjunto variável de números aleatórios. A quantidade exata de números deverá ser informada pelo usuário. Os seguintes algoritmos deverão ser considerados:

- Ordenação bolha (*bubble sort*)
- Ordenação por inserção (*insertion sort*)
- Ordenação por intercalação (*merge sort*)
- Ordenação rápida (*quick sort*)

Após a execução de cada algoritmo, um procedimento de verificação deverá ser executado para garantir que $\text{vetor}[i] \leq \text{vetor}[i+1]$ para qualquer $i < \text{tamanho}(\text{vetor}) - 1$. Em caso de ordenação incorreta, o procedimento deverá informar ao usuário. Note que o programa deverá criar uma cópia do vetor aleatório para cada um dos algoritmos de ordenação.

O programa deverá ainda contar com uma interface com o usuário cuja execução seja semelhante a que segue:

Digite a quantidade de números a serem ordenados: (usuário digita um número X e tecla enter)

Resultado da ordenação de X números:

- 1 – Quick sort (W ms)
- 2 – Merge sort (X ms)
- 3 – Insertion sort (Y ms)
- 4 – Bubble sort (Z ms)

Note que o exemplo de interface anterior apresenta também a classificação dos algoritmos em relação ao tempo utilizado. Obviamente, para um determinado conjunto de dados, a classificação será diferente da apresentada acima, ou seja, é apenas um exemplo. Os tempos W, X, Y e Z representam o tempo de execução do algoritmo em milissegundos ou microssegundos. Como critério adicional de avaliação, poderá ser utilizada a instrução RDTSC (sua utilização é em *assembly inline*) dos processadores x86/x86_64, a qual fornecerá a medição de tempo em nível de ciclos de clock do processador. A combinação de algoritmos de ordenação também apresenta um critério adicional de avaliação.

O que entregar:

1. O código fonte da implementação.
2. Um relatório em PDF, contendo um gráfico XY, cujo eixo X apresente o número de elementos ordenados e Y o tempo gasto por cada algoritmo testado. Nota-se que os valores de X devem ser escolhidos pelos implementadores. Por exemplo, 100, 1000, 10000, 100000, 1000000 são valores válidos para X, sendo que mais valores podem ser considerados.

Obs: O trabalho deverá compilar em qualquer sistema, seja Linux ou Windows. Utilização de funções/includes **não pertencentes a biblioteca padrão** invalidará o trabalho.

Bom trabalho!
