

Exercício Programa 1:

Estudo empírico de algoritmos de ordenação

Rodrigo de Souza

11 de Abril de 2016

1 Introdução

O problema da ordenação consiste em, dado um vetor $v[0..n-1]$ de inteiros ($n > 0$), permutar (trocar de posição) os elementos de forma que o vetor resultante esteja em ordem crescente:

$$v[0] \leq v[1] \leq \dots \leq v[n-1].$$

Para mais detalhes, consulte nosso material didático:

<http://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html>.

O problema da ordenação é um dos mais fundamentais da Computação, e um dos mais importantes, já que ordenação é uma tarefa muito comum, muito requisitada. Convém então que o problema seja resolvido com eficiência.

Desde os primeiros computadores, diversos algoritmos foram propostos para resolver o problema da ordenação. Na verdade, mesmo antes: alguns desses algoritmos são intuitivos, lugar-comum, são procedimentos que utilizamos inconscientemente no dia a dia quando desejamos classificar um fichário, por exemplo. Alguns são mais rápidos do que outros. A finalidade desse exercício é comparar empiricamente alguns algoritmos, na esperança de que você desenvolva um sentimento sobre a diferença de eficiência entre eles (a análise formal desses algoritmos virá depois, na aula).

2 A tarefa

Você deve implementar os seguintes algoritmos de ordenação: inserção, seleção, intercalação (*mergesort*), heapsort, quicksort; todos eles iterativos. Você também deve implementar as versões recursivas de dois desses algoritmos: inserção e intercalação, e no caso do heapsort, fazer duas versões da função peneira, uma iterativa e outra recursiva. A maior parte desses algoritmos você encontra implementada em nosso material: <http://www.ime.usp.br/~pf/algoritmos/>. Neste caso, *é só copiar e colar* (embora se quiser você poderá fazer sua própria versão). Atente bem para o fato de que cada algoritmo corresponde a uma função, que recebe determinados parâmetros bem definidos, e que tem uma documentação dizendo o quê a função faz.

Além de implementar, você deve fazer testes com esses algoritmos, a fim de poder compará-los empiricamente com relação às suas eficiências. Uma sugestão de teste (é o mínimo que você deverá fazer) é gerar vetores (pseudo) aleatórios de diferentes tamanhos, executar cada algoritmo em cada um desses vetores, medir o número de comparações executadas e traçar um gráfico, para cada um deles, onde o eixo das abscissas é o tamanho do vetor e o das ordenadas é o número de comparações. Uma sugestão para gerar vetores aleatórios pode ser encontrada em

<http://www.ime.usp.br/~pf/algoritmos/aulas/random.html#random-permutation>.

Para cada gráfico, você pode tentar inferir a função que se desenha; uma maneira de fazer isso é fazendo uma interpolação, mas se não descobrir como fazer isso, você pode simplesmente comparar os gráficos, tentando classificar os algoritmos com relação à sua eficiência. Você tem liberdade para fazer outros testes que julgar interessantes e relevantes.

Você deve produzir um relatório de laboratório, com uma estrutura sugerida de Introdução, Metodologia, Discussão e Conclusões. Esse relatório deverá ser entregue em pdf juntamente do seu código, que será um único arquivo .c. Você deverá submeter um único arquivo no formato .zip contendo os dois arquivos.

Sugiro que você utilize ferramentas consagradas, livres e de qualidade para produção de material científico. Sugiro que você tente fazer os gráficos usando o Octave ou o Scilab, e o seu relatório em L^AT_EX.

3 Instruções gerais

- Seu programa deve ser feito em C.
- Seu programa deve consistir de um único arquivo, e cada algoritmo corresponde a uma função principal (a que o usuário chama) e dependendo do caso uma função auxiliar. Ou seja, seu código deve consistir das diversas funções implementando os algoritmos pedidos, e da função main, no qual você codifica a bateria de testes a ser analisada no relatório.
- Documente cada função dizendo o quê ela faz.
- Escreva no início do código um cabeçalho com comentários, indicando nome, número do EP, data, nome da disciplina.
- A entrega deverá ser feita no ambiente Moodle, no espaço reservado para essa finalidade. Consulte o ambiente para instruções. A entrega será eletrônica (não receberei exercícios impressos). Você deve enviar um único arquivo compactado contendo seu programa.