Trabalho: 13-ordenacao
Linguagens: PDF
Data de abertura: 2016/11/21 14:00:00
Data limite para envio: 2016/11/28 12:00:00
Número máximo de envios: 25

sqtpm
[183012]

voltar

Ordenação

Implemente algoritmos de ordenação e teste-os com tamanhos variados de vetores aleatórios, registrando os tempos de execução. Depois construa um gráfico com o tamanho das entradas nas abscissas e o tempo nas ordenadas, traçando uma curva do tempo para cada algoritmo.

Sabemos que todos os algoritmos de ordenação imaginados já estão implementados por aí. Mas observe que cada algoritmo representa uma idéia e uma estratégia de solução de problemas em computação. Implementar os algoritmos por si mesmo é uma oportunidade de compreender as idéias, de apreciá-las e de melhorar a técnica pessoal de programação.

Você pode escolher quantos e quais algoritmos implementará e qual a abrangência dos seus testes. Seria interessante incluir também o qsort de stdlib.

Como sugestão, defina os tamanhos das entradas que vai usar e gere alguns vetores de números aleatórios de cada tamanho, depois tire a média dos tempos de execução. Para algoritmos muito lentos, pode ser que o tamanho da entrada tenha que ser pequeno. Isso vai gerar uma curva parcial para o algoritmo, mas não é um problema.

Tome cuidado com entradas que fazem com que um algoritmo exiba comportamento extremo. Vetores que estão ordenados ou quase ordenados são extremos para vários algoritmos de ordenação.

Uma forma de medir o tempo de cpu de um certo trecho de código é envolvendo-o com chamadas a clock_gettime(), subtraindo depois. As definições estão em time.h. Tome cuidado para selecionar o tipo de relógio adequado (algum que meça tempo de CPU do seu processo). Dependendo do ambiente, a resolução do relógio pode ser muito grossa, impedindo a medição adequada para entradas pequenas. Nesse caso pode-se repetir a execução um certo número de vezes, tirando a média depois ou pode-se usar entradas maiores.

Observe que o tempo que será medido não será verdadeiramente médio. Para isso teríamos que medir o tempo de todas as entradas possíveis, mas não é viável.

Números aleatórios podem ser gerados usando rand(). O gerador de

números pode ser inicializado por srand(). Escolhendo a mesma semente k com srand() vai fazer com que rand() produza a mesma seqüência de números, o que permite que os experimentos sejam reproduzidos.

Para gerar o gráfico você pode usar gnuplot, R ou mesmo Excel. Submeta um arquivo pdf de tamanho máximo 1Mb pelo sqtpm.
