MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2001

Análise de tempo de vários algoritmos de ordenação

Devolução: 6 de junho de 2001

Neste projeto você deve implementar alguns algoritmos de ordenação vistos em aula e permitir que o usuário faça alguns testes nos seus algoritmos para verificar a eficiência destes na prática.

O que implementar

Os algoritmos que você deve implementar são os seguintes: *Inserção, Mergesort, Quicksort* e *Heapsort*. As suas implementações devem ser o mais eficientes possíveis e o seu programa deve estar modularizado e bem documentado. Assuma que os dados a serem ordenados são inteiros.

Testes

Quanto aos testes, primeiramente o seu programa deve permitir que o usuário escolha o valor de n, que é o tamanho das seqüências que serão testadas. Além disso, o seu programa deve oferecer ao usuário duas possibilidades de testes: dados aleatórios e dados ordenados em ordem crescente ou decrescente.

No caso de dados aleatórios, o usuário poderá escolher também o número t de testes a serem realizados. O seu programa então deve gerar t seqüências de números escolhidos aleatoriamente entre 1 e n (com possíveis repetições), e ordenar cada uma destas seqüências usando cada um dos algoritmos acima (a mesma seqüência deve ser ordenada por cada um dos algoritmos, ou seja, você precisará fazer uma cópia de cada seqüência).

Estatísticas

Você deve usar o seu programa para montar uma tabela com o tempo de execução de cada um dos diversos algoritmos para $n = 10000, 20000, 40000, 80000, 160000$ e 320000. Sua tabela deve conter os tempos para cada um dos dois tipos de entrada: aleatória e ordenada (crescente e decrescente). A saída deve também ter uma coluna com a razão entre o tempo para cada tipo de entrada e a complexidade do algoritmo (mais precisamente, n, $n \log n$ ou n^2; explicite na tabela ou no relatório por qual destas você dividiu) para cada caso.

No caso de dados aleatórios, o tempo que deve aparecer na tabela deve ser o tempo médio de execução dos testes executados. Para isso, tome $t = 50$.

Para gerar estas estatísticas, você pode usar por exemplo o *gprof*, que é um utilitário que produz algumas estatísticas da execução de um programa. Por exemplo, o *gprof* conta o número de vezes que cada rotina do programa foi executada, quanto tempo foi gasto em cada rotina e qual foi o tempo médio de execução de cada rotina.

Para usar o *gprof*, basta você compilar o seu programa com `gcc -pg` executá-lo e então executar *gprof*. Para mais informações, leia as *man pages* do *gprof* (RTFM ☺).

Relatório e o que você deve entregar

Você deve escrever um relatório (digitado) onde você conta em que máquina ou sistema você executou os seus testes, como você fez as medições de tempo, comenta as medições obtidas, a relação destas com o que foi visto em classe, as suas impressões e eventuais problemas que você tenha tido.

Você deve entregar um disquete contendo os arquivos do seu programa, uma listagem do seu programa, a tabela gerada e o relatório.

Observações

- Certifique-se de que a sua implementação de cada um dos vários algoritmos de ordenação esteja funcionando corretamente.
- Preste atenção na forma de utilizar funções aleatórias no seu programa. O uso inadequado de tais funções pode invalidar os resultados dos experimentos.
- Sinta-se a vontade também de incluir outros algoritmos nos seus testes e outras famílias de instâncias.