

Período: 2009.2
23 de setembro de 2009
Horário: 2as-feiras e 4as-feiras de 15 às 17 horas

ANÁLISE DE ALGORITMOS (INF 1721)

1º Trabalho de Implementação (T1)

Data de Entrega: 23 de novembro de 2009

Descrição

Este trabalho prático consiste em desenvolver códigos para diferentes algoritmos para resolver os problemas descritos abaixo e, principalmente, analisar o desempenho das implementações destes algoritmos com respeito ao tempo de CPU. O desenvolvimento destes códigos e a análise devem seguir os seguintes roteiros:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Explicar a fundamentação do algoritmo e justificar a sua corretude. Apresentar e explicar a complexidade teórica esperada para cada algoritmo.
- Apresentar as tabelas dos tempos de execução obtidos pelos algoritmos sobre as instâncias testadas, comparando sua evolução com a evolução dos tempos seguindo a complexidade teórica correspondente.
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.
- Para a medida de tempo de CPU das execuções utilize as funções disponíveis no link correspondente na página do curso, um exemplo de utilização é apresentado. Quando o tempo de CPU for inferior à 5 segundos, faça uma repetição da execução tantas vezes quantas forem necessárias para que o tempo ultrapasse 5 s (faça um while), conte quantas foram as execuções e reporte a média.

A corretude código será testada sobre um conjunto de instâncias que será distribuído. O trabalho entregue deve conter:

- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos acima, comentários e análises sobre a implementação e os testes realizados (papel).

- A impressão dos trechos relevantes dos códigos fonte (papel).
- Um e-mail para poggi@inf.puc-rio.br (é obrigatório o uso do ASSUNTO (ou SUBJECT) AA092T1 deve ser enviado contendo os arquivos correspondentes ao trabalho. O NÃO ENVIO DESTE E-MAIL IMPLICA QUE O TRABALHO NÃO SERÁ CONSIDERADO).
- O trabalho pode ser feito em grupo de até 3 alunos.

0. Seleção

[K-esimo] Dados um conjunto de n números reais e um inteiro k , encontrar o k -ésimo menor elemento deste conjunto.

1. Implemente o algoritmo da mediana da medianas;
2. Argumente que sua complexidade é linear, i.e. $O(n)$;
3. Verifique sua complexidade experimentalmente nas instâncias disponibilizadas.
4. Analise cuidadosamente, seguindo os passos descritos no início do trabalho, os resultados obtidos.

1. Ordenação

[ORD] Dado um conjunto de n números reais, coloque seus elementos em ordem não-decrescente.

1. Implemente o algoritmo da QuickSort com as seguintes escolhas de pivot:
 - (a) Um elemento aleatório do conjunto considerado na iteração;
 - (b) A média do elementos no conjunto considerado na iteração;
 - (c) A mediana do conjunto considerado na iteração;
2. Implemente o algoritmo Shell Sort. Devem ser implementadas duas versões deste algoritmo. Uma que utilize potência de dois para o cálculo dos intervalos e uma segunda versão que use alguma outra sequência (por exemplo, os números da sequência de Fibonacci (1, 2, 3, 5, 8, 13, ...)). Obtenha as suas respectivas complexidades experimentais. Veja o capítulo 8 do livro Algorithms, R. Sedgewick, 1992.
3. Verifique as complexidades experimentais destes 5 algoritmos nas instâncias disponibilizadas.
4. Analise cuidadosamente, seguindo os passos descritos no início do trabalho, os resultados obtidos.

2. *Problema da Mochila Fracionária (pode-se colocar parte de um objeto na mochila)*

[KP-frac] Dado um conjunto de n objetos divisíveis com pesos positivos w_j , $j = 1, \dots, n$ e valores também positivos v_j , $j = 1, \dots, n$. Sabendo que uma mochila tem a capacidade W , determinar os objetos que podem ser levados na mochila cuja soma dos valores é máxima.

1. Implementar algoritmos para o problema da mochila fracionária com as seguintes complexidades teóricas em função do número n de itens candidatos a serem colocados na mochila:

(a) $O(n \log n)$

(b) $O(n)$

(c) Considere que o seu algoritmo do item (b) utiliza particionamentos em sequência com pivot calculado apropriadamente para garantir a complexidade $O(n)$. Utilize agora como pivot o valor calculado pela expressão:

$$pivot = \frac{1}{|K|} \sum_{j \in K} \frac{v_j}{w_j}$$

onde K é o conjunto de itens considerados.

- i. Prove que a complexidade (pior caso) do algoritmo resultante é $O(n^2)$.
- ii. Estime sua complexidade sobre as instâncias testadas.
- iii. Assim como para os itens (a) e (b) apresente experiências computacionais comparativas.
- iv. Analise cuidadosamente, seguindo os passos descritos no início do trabalho, os resultados obtidos.