

PUC-Rio

Departamento de Informática

Prof. Marcus Vinicius S. Poggi de Aragão

Horário: 2as-feiras e 4as-feiras de 11 às 13 horas - Sala 774L

20 de março de 2006

Data da Entrega: 31 de maio de 2006

Período: 2006.1

PROJETO E ANÁLISE DE ALGORITMOS (INF 2926)

1º Trabalho de Implementação

Descrição

Este trabalho prático consiste em desenvolver códigos para diferentes algoritmos e estruturas de dados 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.
- 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.

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 códigos fonte (papel).
- Um e-mail contendo os códigos fonte e os executáveis correspondentes deve ser enviado para **marcus.poggi@gmail.com**. É obrigatório o uso do ASSUNTO (ou SUBJECT) PAA061T1.
- O trabalho pode ser feito em grupos de até 3 (três) alunos.

0. Estruturas de Dados

Os algoritmos a serem implementados neste trabalho utilizam apenas vetores e matrizes (de acesso direto, i.e. tempo constante) para todas as suas operações.

1. Multiplicação de Polinômios

O grupo deve implementar 3 algoritmos para calcular o polinômio produto de outros dois polinômios. A entrada será dada pelos $n + 1$ coeficientes de cada polinômio de grau n . A saída deverá ser os $2n + 1$ coeficientes do polinômio produto. O algoritmos seguem:

1. Algoritmo que consiste da multiplicação direta dos polinômios de entrada ($O(n^2)$).
2. Algoritmo utilizando divisão-e-conquista ($O(n^{\log_2 3})$).
3. Algoritmo utilizando a DFT e a FFT (*Fast Fourier Transform*) ($O(n \log n)$).

Deve-se determinar qual o algoritmo mais eficiente para todos os valores de n .

2. Problema de Programação Hiperbólica 0-1 Irrestrito (PPH)

- *PPH*: Dado um conjunto de pares ordenados $\{(a_1, b_1), \dots, (a_n, b_n)\}$ e um par obrigatório (a_0, b_0) , onde $a_i, b_i \in \mathbb{Z}^+$, $\forall i = 0, 1, \dots, n$, determinar $S \subseteq N$ onde $N = \{1, \dots, n\}$ que maximiza:

$$R(S) = \frac{a_0 + \sum_{t \in S} a_t}{b_0 + \sum_{t \in S} b_t}$$

Considere o seguinte lema.

Lemma 1 *Seja R^* o valor da razão máxima obtida para o (PPH) e S^* um subconjunto de N tal que $R(S^*) = R^*$. Então, um par t pertence a qualquer S^* se e somente se $a_t/b_t > R^*$.*

Por que o lema acima é verdadeiro ?

Utilize este lema para projetar 3 algoritmos para encontrar R^* e S^* .

1. O primeiro algoritmo inicia com $R = a_0/b_0$ e testa repetidamente se existe algum par (a_k, b_k) que satisfaz às condições do lema. No caso afirmativo, inclui o par no conjunto S , atualiza o valor de R e repete o teste. Observe que se existir um elemento em S que não satisfaz às condições do lema, este elemento deve ser removido.

Este primeiro algoritmo deve executar em $O(n^2)$.

2. Que relação tem o PPH com o problema de ordenação ? Utilize esta observação para projetar um algoritmo de complexidade $O(n \log n)$
3. Observe novamente a relação do PPH com o problema de ordenação. O que caracteriza o conjunto S^* ? Esta caracterização permite projetar um algoritmo de complexidade $O(n)$. Apresente este algoritmo.

Novamente implemente os algoritmos acima e determine qual o mais eficiente para cada instância executada. Indique para que faixas de valores de n cada algoritmo é o mais eficiente.