PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão (3WA) Lorenza Moreno e David Sotelo (3WB)
Horário: 2as. e 4as. 15-17hs (3WA), 3as. e 5as. 19-21 (3WB)
31 de março de 2010
Data da Entrega: 22 de maio de 2010
Período: 2010.1

ANÁLISE DE ALGORITMOS (INF 1721)

1^o 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 experimental 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 distribuido. 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 **poggi@inf.puc-rio.br**. É **OBRIGATÓRIO** o uso do ASSUNTO (ou SUBJECT) AA101T1, **a falta do e-mail COM este ASSUNTO implica na NÃO consideração do trabalho**.
- O trabalho pode ser feito em grupos de até 2 (dois) 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. Controle de Qualidade na Produção de Frascos de Vidro

O controle de qualidade de uma fábrica de frascos de vidro precisa determinar o maior nível de impacto que os seus frascos podem receber sem quebrar. Para este fim, a seção de controle de qualidade possui uma longa escala de alta precisão postada verticalmente. O valor na escala (x) a partir do qual o frasco quebra é o que determina a resistência de cada tipo de produto oferecido pela fábrica. Considere que esta escala possui comprimento total n. Se para determinar x, onde $1 \leq x \leq n$, somente um frasco está disponível, o único algoritmo que garante a determinação de x consiste em provocar a queda do frasco do valor 1, depois do 2, e assim por diante até o frasco quebrar com a sua queda. O número de quedas provocadas neste algoritmo é $O(n)$.

O grupo deve projetar e implementar algoritmos que simulem cada queda provocada até a determinação do valor em que o frasco quebra. A entrada do algoritmos é o valor de x (a resposta!) que será usado para determinar se o frasco vai quebrar ou não com a sua queda. O algoritmo deve executar até provar que a altura em que o frasco quebra é x.

Suponha agora que para um tipo de frasco, exista mais de um frasco disponível para determinar o valor a partir do qual quebrará. Examine os casos abaixo, proponha e implemente os algoritmos para "determinar" o valor x.

1. Considere que 2 frascos estão disponíveis. Proponha para este caso um algoritmo onde o número de quedas provocadas dos frascos é $O(\sqrt{n})$ e que a determinação do valor x é garantida.

2. Mostre que para k frascos disponíveis é possível propor uma algoritmo onde o número de quedas é, no pior caso, $k.n^{\frac{1}{k}}$. Apresente os algoritmos para $k = 3$ e $k = 4$. Em seguida generalize.

3. Qual a menor complexidade assintótica possível para um algoritmo determinar o valor de x ? Proponha um algoritmo que tenha essa complexidade. Quantos frascos são necessários para que esta complexidade seja atingida ?

O grupo deve implementar e testar experimentalmente 5 algoritmos (para 1, 2, 3 e 4 frascos disponíveis e para um número ilimitado deles) apresentando estatísticas do tempo de CPU das execuções para cada conjunto de valores de x. Estes serão correspondentes ao número de bits utilizados para representar o valor máximo do conjunto (n é este valor máximo). Haverá conjuntos de 32, 64, 128, 192 e 256 bits. Estas instâncias do problema serão disponibilizadas na página *web* do curso.

- *PPH*: Dado um conjunto de pares ordenados $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ e um par obrigatório (a_0, b_0), onde $a_i, b_i \in Z^+, \quad \forall\, i = 0, 1, \ldots, n$, determinar $S \subseteq N$ onde $N = \{1, \ldots, n\}$ 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 4 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.

   Para este algoritmo, apresente sua complexidade assintótica e a análise correspondente.

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.

4. Implemente o seguinte algoritmo para o PPH:

Passo 0 : Inicialização: Faça $S = N$ e calcule $R(S)$.

Passo 1 : Seja M o conjunto do elementos de S onde a razão associada (a/b) é menor que $R(S)$. Se M é vazio, PARE, S é a solução ótima do PPH. Caso contrário, remova os elementos em M de S e repita o passo 1.

Apresente e analise a complexidade assintótica deste 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. Instâncias para este problema serão disponibilizadas na página do curso.