PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2009.1
28 de abril de 2009
Horário: 3as-feiras e 5as-feiras de 17 às 19 horas

Estruturas Discretas (INF 1631)

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 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 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 trechos relevantes dos códigos fonte (papel).
- Um e-mail para poggi@inf.puc-rio.br (é obrigatório o uso do ASSUNTO (ou SUBJECT) ED091T1 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.

1. Considere o teorema abaixo e a sua prova.

Teorema 1 *: $x^n - y^n$ é divisível por $x - y$ para quaisquer x e y inteiros e todos o valores de n inteiros e maiores que zero.*

Prova 1 *A prova é feita por indução matemática utilizando k como parâmetro de indução. O teorema 1 pode ser enunciado:*

Teorema 1 *(k): $x^k - y^k$ é divisível por $x - y$ para quaisquer x e y inteiros e todos o valores de k inteiros e maiores que zero.*

Teorema do Caso Base: 1 *Seja $k = 1$ (o menor valor para o qual k tem que ser verdade). Nesse caso temos que provar que $x - y$ é divisível por $x - y$ para qualquer valor de x e y. O que é trivialmente verdade, sendo o quociente, q_1, igual a 1.*

Teorema do Passo Indutivo: 1 *Desejamos provar que se o teorema 1 é verdade para um k fixo, isto é, podemos assumir que:*

$$x^k - y^k = q_k.(x - y)$$

onde q_k é um inteiro, então é possivel mostrar que

$$x^{k+1} - y^{k+1} = q_{k+1}.(x - y)$$

para q_{k+1} inteiro. Ou seja, temos que mostrar é verdade também para $k + 1$. Isto é, que podemos obter q_{k+1} inteiro a partir de q_k se o teorema 1 é verdade para k.

Como:

$$x^{k+1} - y^{k+1} = x^{k+1} - x^k.y + x^k.y - y^{k+1} = x^k(x - y) + y(x^k - y^k)$$

Como, pela hipótese indutiva, temos que $x^k - y^k = q_k.(x - y)$, podemos escrever:

$$x^{k+1} - y^{k+1} = x^k(x - y) + y(x^k - y^k) = x^k(x - y) + y.\mathbf{q_k}.(\mathbf{x} - \mathbf{y}) = (x^k + y.q_k)(x - y)$$

Como x é inteiro, x^k é inteiro. Como y e q_k são inteiros seu produto também é inteiro. Portanto $(x^k + y.q_k)$ é inteiro e $q_{k+1} = (x^k + y.q_k)$ é inteiro, ou seja:

$$x^{k+1} - y^{k+1} = (x^k + y.q_k)(x - y) = q_{k+1}.(x - y)$$

O teorema acima resolve o problema de determinar o quociente entre $x^k - y^k$ e $x - y$. Assim deseja-se um algoritmo que dados x, y, e k inteiros determine esse quociente.

(a) Escreva o algoritmo resultante da prova acima.

(b) Implemente este algoritmo e teste para vários valores de x, y, e k.

2. Considere o teorema abaixo:

   Teorema 2 *o número de números inteiros cujos dígitos pertencem ao conjunto* $\{1, 2, \ldots, m\}$ *de* K *dígitos diferentes é dado pelo produto* $m.(m-1)\ldots..(m-k+1)$.

   (a) Enuncie o teorema de que sabe-se enumerar todos estes números especificando seu parâmetro indutivo e prove-o por indução matemática (simples).

   (b) Apresente o algoritmo resultante da sua prova, que enumera todos os $m.(m-1)\ldots..(m-k+1)$ números (o que permite contá-los).

   (c) Implemente este algoritmo e apresente os números inteiros (a sequência de dígitos, em especial para quando m é maior que nove) impressos para pequenos valores de n e m. Para valores maiores apresente o tempo de CPU e indique até que valores de m e k sua implementação (e computador) foi capaz de fazer a enumeração.

3. Seja um grafo conexo $G = (V, A)$, orientado, e ℓ_a, para $a \in A$ os pesos associados aos arcos de G. Seja o problema de encontrar o caminho mais curto (de menor peso nesse caso) entre todos os pares de vértices em V. Sejam os teoremas abaixo.

   Teorema 3 *(K): Sabe-se encontrar o caminho mais curto entre* u *e* w $\forall u, w \in V$ *(ou seja, entre todos os pares de vértices em* V*) que utiliza no máximo* K *arcos.*

   Teorema 4 *(K): Sabe-se encontrar o caminho mais curto entre* u *e* w $\forall u, w \in V$ *(ou seja, entre todos os pares de vértices em* V*) que utiliza como vértices intermediários o conjunto* $\{1, 2, \ldots, K\}$.

   (a) Apresente a prova por indução matemática no parâmetro K (número máximo de arcos no caminho) do Teorema 3;

   (b) Apresente o algoritmo correspondente à prova do Teorema 3 apresentada e a sua respectiva implementação.

   (c) Apresente a prova por indução matemática no parâmetro K (índice máximo do vértice que pode estar no caminho de cada para de vértices) do Teorema 4;

   (d) Apresente o algoritmo correspondente à prova do Teorema 4 apresentada e a sua respectiva implementação.

   (e) Teste os algoritmos nas instâncias de caminho mais curto disponibilizadas na página do curso.