

PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2014.1
16 de maio de 2014
Horário: 2as-feiras e 4as-feiras de 11 às 13 horas

Estruturas Discretas (INF 1631)

2º Trabalho de Implementação

Data de Entrega: 11 de junho de 2014 - quarta-feira

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 correteza apresentando a prova por indução matemática que leva ao algoritmo.
- Apresentar as tabelas dos tempos de execução obtidos pelos algoritmos sobre as instâncias testadas.
- 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 correteza código deverá ser testada sobre um conjunto de instâncias. 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 (somente) dos trechos relevantes dos códigos fonte (papel).
- Um e-mail para poggi@inf.puc-rio.br (é obrigatório o uso do ASSUNTO (ou SUBJECT) ED141T2 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. Seja um grafo conexo $G = (V, E)$, não-orientado, e w_e , para $e \in E$ os pesos associados às arestas de G . Seja o problema de encontrar a árvore geradora de peso máximo de G que contém o vértice $1 \in V$ e o teorema abaixo.

Teorema 1 (K): *Sabe-se encontrar a árvore de peso máximo de $G = (V, E)$ que contém o vértice 1 e possui K vértices.*

- (a) Prove este teorema por indução simples.
- (b) Apresente o algoritmo correspondente.
- (c) Escreva seu código explicitando os trechos do mesmo que contém a prova (cada um dos seus passos) do caso base e do passo indutivo, a prova garante a corretude do seu algoritmo.
- (d) Teste seu algoritmo em instâncias pequenas dos arquivos de dados fornecidos na página do curso (os formatos dos arquivos das instâncias estão incluídos). Teste em uma instância pequena, desenhe o grafo e mostre os passos do seu algoritmo e que o seu código encontrou uma solução correta.
- (e) DESAFIO (1.0 ponto extra no grau G2)
Repita os itens anteriores para o teorema abaixo:

Teorema 2 (K): *Sabe-se encontrar a floresta de peso mínimo de $G = (V, E)$ onde os componentes conexos possuem pelo menos K vértices.*

2. *Walk of a King*

Considere um tabuleiro de xadrez e um rei que está inicialmente na posição $(1, 1)$ (as posições do tabuleiro são representadas por (i, j) onde $1 \leq i \leq 8$ e $1 \leq j \leq 8$). Para cada posição do tabuleiro estão associados um prêmio p_{ij} e um consumo q_{ij} (o prêmio pode ser em USD(!!) e o consumo em litros de gasolina, por exemplo). Os prêmios e os consumos assumem somente valores positivos. O rei tem inicialmente Q unidades para consumir e pode passar quantas vezes quiser em cada posição do tabuleiro e a cada vez receber o prêmio e, naturalmente, consumir os seus recursos. Ao final (do passeio) **o rei tem que estar de volta na posição $(1, 1)$.**

(a) Prove o teorema a seguir por indução:

Teorema 3 (i, j, q) : *Sabe-se determinar o prêmio máximo que o rei consegue coletar estando na posição (i, j) com q unidades restantes para consumir.*

- (b) Com base na prova do teorema do item anterior, apresente um algoritmo para determinar o caminho que o rei deve fazer para obter o maior total possível em prêmios.
- (c) Desenvolva um código para o seu algoritmo e teste para um arquivo de dados com a estrutura apresentada abaixo.

Descrição do Arquivo de Entrada

A primeira linha contém um inteiro Q que indica o número de unidades que o rei tem para consumir. Seguem dois blocos de 8 linhas cada uma com 8 inteiros positivos. O primeiro bloco corresponde ao consumo de cada posição do tabuleiro, enquanto que o segundo bloco corresponde ao prêmios atribuídos às posições.

A posição $(1, 1)$ é a primeira das oito na primeira linha de cada bloco. Na mesma linha se encontram as posições $(1, 2)$, $(1, 3)$, \dots $(1, 8)$. A linha seguinte se inicia com o valor da posição $(2, 1)$, a linha seguinte com o valor da posição $(3, 1)$ e assim por diante. A última linha de cada bloco contém valores referentes às posições $(8, 1)$, $(8, 2)$, \dots , $(8, 8)$. Naturalmente esta linha é precedida pela linha $(7, 1)$, $(7, 2)$, \dots , $(7, 8)$.

O arquivo de dados possui múltiplas instâncias e termina quando o valor de Q é zero.

Descrição do Arquivo de Saída

Para cada entrada do arquivo acima você deve imprimir o maior prêmio possível, o total consumido, o valor de Q e a sequência de posições correspondente à caminhada do rei.