

Período: 2008.1

15 de abril de 2008

Horário: 2as-feiras e 4as-feiras de 9 às 11 horas

ANÁLISE DE ALGORITMOS (INF 1721)

1º Trabalho de Implementação (T1b) (Parte b)

Data de Entrega: 23 de junho de 2008

Descrição

Este trabalho prático consiste em desenvolver códigos para 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 os conjuntos de instâncias distribuídos. 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). **É parte importante do documento uma tabela contendo os tempos de CPU de cada um dos algoritmos pedidos (descritos abaixo) para cada instância dada.**
- Um e-mail para poggi@inf.puc-rio.br deve ser enviado contendo os códigos fonte, os executáveis correspondentes e o documento final (é obrigatório o uso do ASSUNTO (ou SUBJECT) AA081T1b). (Lembrem que o gmail não aceita .exe, e portanto vc deve modificar a terminação de .zip para .txt).
- O trabalho pode ser feito em grupo de até 2 alunos.

0. Estruturas de Dados

O grupo deve implementar (ou usar códigos prontos) códigos para efetuar as seguintes operações nas estruturas de dados abaixo:

1. *d-Heap* para um valor de d qualquer.

1. *Problema da Caminho-mais-curto em grafos com distâncias não-negativas*(veja o link “*Instâncias para CMC*” para grafos de teste)

1. Implementar o Algoritmo de Dijkstra utilizando as estruturas de dados, listadas a seguir, para selecionar o próximo vértice mais próximo. Nestas estruturas, cada vértice tem como valor-chave a distância do menor caminho corrente que o conecta ao vértice de partida.

Lista de estruturas de dados a utilizar:

(a) *Vetor* com os valores das distâncias.

(b) *d-Heap*, para os valores $d = 2$, $d = 3$ e $d = 5$.

2. *Problema da Fluxo-Máximo em grafos* (veja o link “*Instâncias para Fluxo Máximo*” para grafos de teste)

1. Implementar o Algoritmo de Ford-Fulkerson (usa Busca em Profundidade para encontrar um caminho s-t na rede residual);

2. Implementar o Algoritmo de Edmonds-Karp (usa Busca em Largura para encontrar um caminho s-t na rede residual);