

PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão (3WA)
Horário: 2as. e 4as. 9-11hs (3WA)
16 de setembro de 2015
Data da Entrega: 12 de outubro de 2015
Período: 2015.2

ANÁLISE DE ALGORITMOS (INF 1721)

2º 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 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 e digital).
- Código fonte (só digital, todos os arquivos para gerar o executável).
- 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) **AA152T2, a falta do e-mail COM este ASSUNTO implica na NÃO consideração do trabalho e nota Zero!**
- O trabalho pode ser feito em grupos de até 3 (três) alunos.

Algoritmos de Fluxo Máximo e sua Aplicação em Classificação de Dados

O objetivo do trabalho é implementar e testar dois algoritmos básicos para a determinação do fluxo máximo entre pares de vértices: Edmonds-Karp e Push-Relabel, descritos abaixo. Como teste, estes algoritmos devem ser aplicados em grafos para determinar uma separação dos vértices do grafo em K componentes (conjuntos de vértices). A descrição detalhada dos algoritmos, da aplicação e das instâncias a serem resolvidas seguem abaixo.

Como no trabalho anterior, códigos prontos podem ser utilizados desde que todo o fonte seja fornecido na entrega do trabalho e a documentação indique a origem do código e o que foi e não foi modificado pelo grupo.

- Considere o problema de determinar o Fluxo Máximo de um vértice s a um vértice t em um grafo orientado $G = (V, E)$ onde os arcos $e \in E$ têm capacidade igual c_e , $|V| = n$ e $|E| = m$. Lembre que uma rede residual associada a um fluxo $f = (f_e, e \in E)$ possui para cada arco $e = (u, v)$ um arco (u, v) com capacidade $c_e - f_e$ se $c_e - f_e > 0$ e um arco (v, u) com capacidade f_e se $f_e > 0$. Considere agora o algoritmo para encontrar o fluxo máximo de s para t em um grafo, abaixo:

1. Algoritmo de Edmonds-Karp

Algoritmo EdKarp (s - fonte, t - sumidouro)

Passo 0: *Inicialização*

Seja $f = (f_1, \dots, f_m)$ o fluxo corrente nos arcos de E , $|E| = m$. Seja $R(f)$ a rede residual do grafo $G = (V, E)$ dado o fluxo corrente f .

$f \leftarrow (0, \dots, 0)$

Passo 1: *Iteração*

Enquanto existe caminho de s a t em $R(f)$ faça

- 1.1 Encontre o caminho mais curto (em número de arcos) de s a t em $R(f)$. Seja p_1, \dots, p_k esse caminho (que tem k arcos).
- 1.2 Seja $q = \min_{e=p_1, \dots, p_k} r(e)$ onde $r(e)$ é a capacidade do arco e na rede residual $R(f)$.
- 1.3 Atualização de f $R(f)$: Para $e = p_1, \dots, p_k$ faça $f_e \leftarrow f_e + q$; Atualize $R(f)$.

2. Algoritmo Preflow-Push (Karzanov)

Um *pré-fluxo* é uma atribuição de valor aos fluxos nos arcos, valores das variáveis f_e para $e \in E$, onde para todo vértice v a soma dos fluxos que entram é maior ou igual à soma dos fluxos que saem. Em um *fluxo* temos sempre a igualdade.

Algoritmo Push-Relabel (s - fonte, t - sumidouro) Seja $f = (f_1, \dots, f_m)$ o fluxo corrente nos arcos de E , $|E| = m$.

Seja $R(f)$ a rede residual do grafo $G = (V, E)$ dado o fluxo corrente f .

Seja $d(v)$ a distância de v até o vértice sumidouro t (# arcos)

Seja $e(i)$ o excesso de fluxo do vértice i :
$$e(i) = \sum_{(j,i) \in E} f_{(j,i)} - \sum_{(i,j) \in E} f_{(i,j)}$$

Preprocess

- *Inicialização*

$f \leftarrow (0, \dots, 0)$

Calcule as distâncias $d(v)$ para $v \in V$

Faça $f_{(s,u)} = c_{(s,u)}$ para $(s,u) \in E$

$d(s) \leftarrow n$

Push/Relabel (i) - i um vértice ativo de V (i.e. $e(i) > 0$)

- *Iteração*

Se existe um arco **admissível** (i, j) em $R(f)$:

1.1 **Push**: faça $f_{(i,j)} = f_{(i,j)} + \delta$ onde $\delta = \min\{e(i), r((i,j))\}$

Observe que se (i, j) for um arco reverso, δ será subtraído de $f_{(j,i)}$

Caso contrário (else):

1.2 **Relabel**: faça $d(i) = \min\{d(j) + 1, \text{ para } (i, j) \in R(f)\}$.

Preflow Push - (s - fonte, t - sumidouro)

- **Preprocess**

- Enquanto existe um vértice i ativo (I.e., i tal que $e(i) > 0$)

Escolha um vértice ativo i

Push/Relabel(i)

Aplicação em Classificação de Dados: resolva problema a seguir

- Dado um conjunto de objetos e os valores das conexões par a par, quanto maior, mais conectados são os objetos, separar os objetos em K conjuntos de modos que estes estejam o mais separados possível.
 - O algoritmo de resolução deste problema consiste em encontrar, sucessivamente, o par de (grupos de) vértices cujo fluxo máximo (que corresponde ao corte mínimo), é o menor possível.
 - Ou seja, após encontrar o par de objetos entre os quais o fluxo máximo é mínimo, o conjunto de vértices (objetos) é dividido em dois. Cada um destes conjuntos será dividido em dois da mesma forma até que sejam formados K conjuntos. Observe que a ordem das divisões será pela ordem crescente do valor do fluxo máximo (corte mínimo) entre o par que divide o conjunto corrente.
- Aplique este algoritmo às instâncias fornecidas (10). Para cada uma resolva K assumindo valores: 3, 4 e 5. Reporte os tempos de execução de cada resolução do problema de Fluxo Máximo para os dois algoritmos descritos acima.
- Analise os resultados obtidos quanto à performance dos algoritmos de fluxo máximo e ao que se pode interpretar dos resultados em termos de classificação.