

# Algoritmos em Grafos

## Terceiro Trabalho Prático

### 1 Descrição Geral

Neste terceiro trabalho vocês deverão implementar os algoritmos de fluxos de Ford-Fulkerson e Edmonds-Karp, bem como estruturas de dados associadas, e fazer uma comparação entre os tempos de execução dos algoritmos entre si. No algoritmo de Ford-Fulkerson use algum outro algoritmo que não a BFS para encontrar um caminho aumentante, por exemplo uma DFS.

O trabalho deverá ser feito **pela mesma equipe do trabalho I**. O trabalho deve ser implementado em uma das seguintes linguagens: C/C++, Java ou Python 3 (não será aceito Python 2).

Todos os dados da sua implementação devem estar dentro de uma pasta chamada **fluxos** e dentro desta pasta deverá haver uma pasta **sources** contendo os códigos fontes das implementações dos algoritmos. Na pasta **fluxos** deverá haver um arquivo Makefile que será executado em ambiente Linux/Unix para geração de um executável que deverá se chamar **main ou main.jar** (o executável main deverá ser criado dentro da pasta **fluxos**). No caso de python deverá ser criado um arquivo main.py na pasta **fluxos** sem necessidade de um arquivo Makefile.

A execução de main receberá 3 parâmetros pela linha de comando:

1. **ford-fulkerson** ou **edmonds-karp**: indicando qual algoritmo será executado.
2. **arq-in**: um caminho para um arquivo com instância de entrada.
3. **arq-out**: um caminho para geração de um arquivo de saída.

### 2 Formato de Arquivos

Os arquivos de entrada devem seguir o seguinte padrão:

```
n
m
s
t
v_1 v_1' c(v_1, v_1')
.
.
.
v_m v_m' c(v_m, v_m')
```

onde  $n$  é um inteiro indicando o número de vértices no grafo e  $m$  é outro inteiro indicando o número de arestas no grafo,  $s$  é um inteiro em  $[0, n - 1]$  indicando

o vértice fonte, e  $t \neq s$  é um inteiro em  $[0, n - 1]$  indicando o vértice destino. Nas próximas  $m$  linhas temos as informações de cada aresta: os dois primeiros valores, separados por 1 espaço, são índices em  $[0, n - 1]$  indicando os vértices incidentes na aresta e depois de mais 1 espaço, um número tipo **int** indicando a capacidade desta aresta.

Os arquivos de saída devem ser gerados da seguinte forma:

```
v_1 v_1' f(v_1, v_1')
.
.
.
v_m v_m' f(v_m, v_m')
```

onde cada linha tem a especificação de uma aresta. A saída tem exatamente as mesmas arestas do arquivo de entrada e na mesma ordem, mas agora o terceiro valor é um **int** indicando o fluxo enviado por aquela aresta.

### 3 Geração de Instâncias

Vocês poderão usar instâncias aleatórias, mas sugerimos fortemente o uso de instâncias padrões de outros problemas em grafos.

Na página <http://www.math.uwaterloo.ca/tsp/world/countries.html> existem instâncias geradas a partir de mapas de países.

Já na página <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/> existem instâncias usadas para o problema TSP.

O formato destas instâncias está explicado aqui <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp95.pdf>.

Considere apenas o formato de instâncias para TSP simétrico (basicamente são dadas coordenadas no plano dos vértices e as distâncias entre vértices correspondem a distância euclidiana entre os pontos).

Se utilizar tais instâncias transforme-as para o formato padrão deste trabalho.

### 4 Informações Sobre Envio

Até o dia 10/Junho/2019 cada grupo deverá enviar um email para [eduardo@ic.unicamp.br](mailto:eduardo@ic.unicamp.br) contendo o trabalho seguindo estritamente as seguintes especificações (caso contrário o trabalho não será considerado):

- O *subject* da mensagem deve conter apenas: mo412 fluxos
- No corpo da mensagem inclua os nomes e RAs dos alunos participantes do grupo.
- Anexado à mensagem deve haver um único arquivo **fluxos.zip** que quando descompactado gerará uma pasta **fluxos-ra1-ra2-ra3** onde os campos **ra1** até **ra3** deverão conter os RAs dos alunos do grupo separados por

um hífen. Dentro da pasta **fluxos-ra1-ra2-ra3** deverá haver outra pasta **fluxos** com a estrutura especificada na Seção 1.

- Os trabalhos devem ser enviados até o dia 10/Junho/2019 às 11:59pm.

No final do semestre cada grupo deverá fazer uma apresentação de 30 minutos em sala com os resultados de suas implementações. Nesta apresentação deverão ser apresentados os resultados de todos os trabalhos.