

Algoritmos em Grafos

Primeiro Trabalho Prático

1 Descrição Geral

Neste primeiro trabalho vocês deverão implementar os algoritmos de Prim e Kruskal, bem como estruturas de dados associadas, e fazer uma comparação entre os tempos de execução dos algoritmos entre si, bem como uma comparação com o uso de diferentes estruturas de dados para manutenção do heap e estruturas de union-find.

O trabalho deverá ser feito em equipe, composta por 2 até 3 alunos. 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 **mst** e dentro desta pasta deverá haver uma pasta **sources** contendo os códigos fontes das implementações dos algoritmos. Na pasta **mst** 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 **mst**). No caso de python deverá ser criado um arquivo main.py na pasta **mst** sem necessidade de um arquivo Makefile.

A execução de main receberá 4 parâmetros pela linha de comando:

1. **prim** ou **kruskal**: indicando qual algoritmo será executado.
2. **x**: onde x é um inteiro indicando qual versão do algoritmo de prim ou kruskal utilizar, onde uma versão é uma implementação específica com uma determinada estrutura de dados.
3. **arq-in**: um caminho para um arquivo com instância de entrada.
4. **arq-out**: um caminho para geração de um arquivo de saída.

Dentro da pasta **mst** crie um arquivo LEIAME.txt onde você deverá informar o número máximo de opções **x** para o seu algoritmo Prim e Kruskal. A primeira opção deve ser necessariamente o número 0. Para cada opção indique o inteiro que deve ser usado e indique também quais estruturas de dados foram usadas na implementação desta opção específica.

2 Formato de Arquivos

Os arquivos de entrada devem seguir o seguinte padrão:

```
n
m
v_1 v_1' w(v_1, v_1')
.
.
```

```
.
v_m v_m' w(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. 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 **float** indicando o peso desta aresta.

Os arquivos de saída devem ser gerados da seguinte forma:

```
custo-sol
v_1 v_1' w(v_1, v_1')
.
.
.
v_{n-1} v_{n-1}' w(v_{n-1}, v_{n-1}')
```

onde na primeira linha devemos ter o custo da MST encontrada e em seguida em cada linha a indicação das arestas usadas na solução. Em cada linha temos os índices em $[0, n - 1]$ dos vértices que compõem a aresta, separados por 1 espaço, e depois de 1 espaço deve-se ter o peso da aresta. Notem que devem existir exatamente $n - 1$ arestas.

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 15/Maio/2019 cada grupo deverá enviar um email para 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 mst

- No corpo da mensagem inclua os nomes e RAs dos alunos participantes do grupo.
- Anexado à mensagem deve haver um único arquivo **mst.zip** que quando descompactado gerará uma pasta **mst-ra1-ra2-ra3** onde os campos **ra1** até **ra3** deverão conter os RAs dos alunos do grupo separados por um hífen. Dentro da pasta **mst-ra1-ra2-ra3** deverá haver outra pasta **mst** com a estrutura especificada na Seção 1.
- Os trabalhos devem ser enviados até o dia 15/Maio/2019 às 11:59pm.

No final do semestre cada grupo deverá fazer uma apresentação de 25 minutos em sala com os resultados de suas implementações. Nesta apresentação deverão ser apresentados os resultados deste trabalho e de outros 2 trabalhos a serem especificados posteriormente.