

Teoria dos Grafos – COS 242

2020/2

Trabalho de Disciplina – Parte 2

1 Logística

Esta é a segunda parte do trabalho da disciplina. Você deve incorporar este trabalho na biblioteca implementada na primeira parte. Se você fez a primeira parte em dupla, então a dupla deve continuar a mesma. Como na primeira parte, seu relatório deve informar as decisões de projeto e de implementação das funcionalidades abaixo, responder às perguntas relacionadas aos estudos de caso, e conter no **máximo 4 páginas**.

2 Descrição

Funcionalidades que precisam ser implementadas pela sua biblioteca:

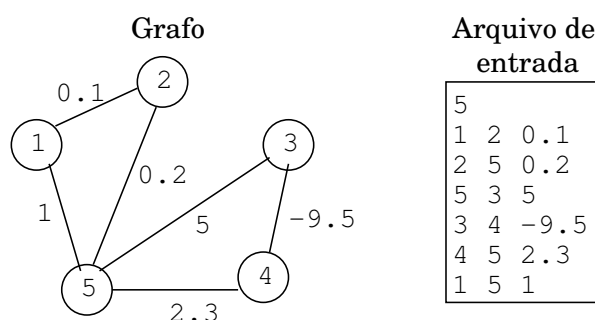


Figura 1: Exemplo de grafo com pesos e o formato do arquivo.

1. **Grafos com pesos.** Sua biblioteca deve ser capaz de representar e manipular grafos não-direcionados que possuam pesos nas arestas. Os pesos, que serão representados por valores reais, devem estar associados às arestas. Você deve decidir a melhor forma de estender sua biblioteca de forma a implementar esta nova funcionalidade. O arquivo de entrada será modificado, tendo agora uma terceira coluna, que representa o peso da aresta (podendo ser qualquer número de ponto flutuante). Um exemplo de um grafo não-direcionado com pesos e seu respectivo arquivo de entrada está ilustrado na figura 1.
2. **Distância e caminho mínimo.** Sua biblioteca deve ser capaz de encontrar a distância entre qualquer par de vértices assim como um caminho que possui esta distância. Se o grafo não possuir pesos, o algoritmo de busca em largura deve ser utilizado. Se o grafo possuir pesos, o algoritmo de Dijkstra deve ser utilizado. Neste último caso, é necessário verificar se os pesos de todas as arestas são maiores ou iguais a zero, condição necessária para que o algoritmo de Dijkstra funcione corretamente.
Você deve decidir como implementar o algoritmo de Dijkstra em sua biblioteca (por exemplo, usando um *heap* binário), lembrando que isto irá influenciar o tempo de execução do seu algoritmo. Além de calcular a distância e caminho mínimo entre um par de vértices, sua biblioteca deve ser capaz de calcular a distância e caminho mínimo entre um dado vértice e todos os outros vértices do grafo.
3. **Árvore geradora mínima (MST).** Sua biblioteca deve ser capaz de encontrar uma árvore geradora mínima de um grafo. Você deve escolher um algoritmo apropriado para

resolver este problema. A árvore geradora mínima deve ser escrita em um arquivo (no mesmo formato que um grafo), assim como seu peso total.

4. **Excentricidade.** A excentricidade de um vértice u é a maior distância entre o vértice e qualquer outro vértice do grafo, ou seja, $c_u = \max_v d(u, v)$, onde $d(u, v)$ é distância entre os vértices u e v . Sua biblioteca deve calcular a excentricidade de um vértice qualquer.

3 Estudos de Caso

Considere os grafos com pesos disponíveis no website da disciplina. Para cada grafo, responda às perguntas abaixo.

1. Calcule a distância e o caminho mínimo entre o vértice 1 e os vértices 10, 20, 30, 40, 50.
2. Calcule a excentricidade dos vértices 10, 20, 30, 40, 50.
3. Determine o tempo médio para calcular a excentricidade. Ou seja, escolha k vértices iniciais (ex. $k = 100$) de forma aleatória e calcule a excentricidade deles marcando o tempo e fazendo a média amostral.
4. Obtenha uma árvore geradora mínima, informando seu peso. Obtenha o tempo de execução para resolver este problema.

Considere a rede de colaboração entre pesquisadores da área de Computação disponível no website da disciplina, onde o peso da aresta indica proximidade entre os pesquisadores (peso é inversamente proporcional ao número de artigos publicados em co-autoria). Responda às perguntas abaixo.

1. Calcule a distância e o caminho mínimo entre Edsger W. Dijkstra (o pesquisador) e os seguintes pesquisadores na rede de colaboração: Alan M. Turing, J. B. Kruskal, Jon M. Kleinberg, Éva Tardos, Daniel R. Figueiredo. Utilize exatamente estes nomes (string) para identificar os índices dos vértices.