

Teoria dos Grafos – COS 242

2021/2

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 5 páginas**. O relatório deve conter a URL para o código-fonte da sua implementação. A dupla deve preparar uma apresentação em video de no máximo 5 minutos sobre o trabalho. O video deve ser disponibilizado na nuvem (ex. Google Drive da Poli) e a URL submetida no Moodle.

2 Descrição

Funcionalidades que precisam ser implementadas pela sua biblioteca:

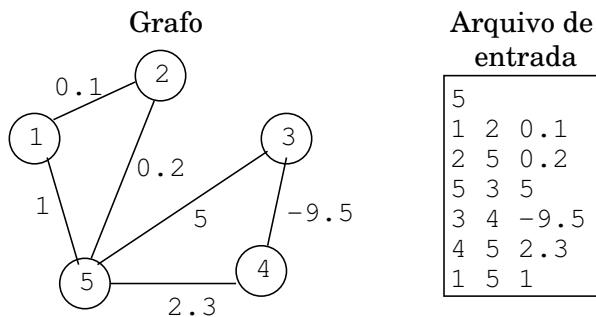


Figura 1: Exemplo de grafo com pesos e o formato do arquivo.

- 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.
- 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 não negativos, o algoritmo de Dijkstra deve ser utilizado. Se o grafo possuir pesos negativos, o algoritmo de Floyd-Warshall ou o algoritmo de Bellman-Ford devem ser utilizados. Neste caso, a biblioteca deve também informar se o grafo possui ciclos negativos (caso no qual as distâncias não estão definidas).

Você deve decidir como implementar os algoritmos em sua biblioteca. Por exemplo, usando um *heap* binário para o algoritmo de Dijkstra. Lembrando que isto irá influenciar o tempo de execução e consumo de memória da sua implementação.

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.

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. Determine o tempo médio para calcular a distância entre um vértice e todos os outros vértices do grafo. Escolha k vértices iniciais (ex. $k = 100$) de forma aleatória, calcule o tempo de cada execução, e faça a média amostral.
3. Obtenha uma árvore geradora mínima, informando seu peso.

Considere a rede de colaboração entre pesquisadores da área de Computação disponível no website da disciplina, onde o peso da aresta indica a 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 da rede de colaboração: Alan M. Turing, J. B. Kruskal, Jon M. Kleinberg, Éva Tardos, Daniel R. Figueiredo. Utilize exatamente estes nomes (strings) para identificar os índices dos vértices.