Teoria dos Grafos – COS 242

2022/2

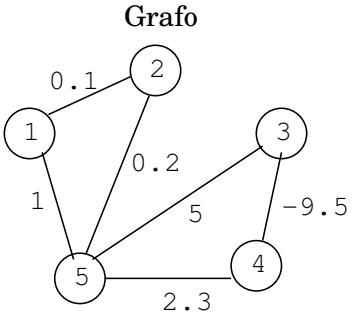
Trabalho de Disciplina – Parte 2

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 para ser feita em aula de no máximo 8 minutos sobre o trabalho.

2 Descrição

Funcionalidades que precisam ser implementadas pela sua biblioteca:



Arquivo de entrada

```
5
1 2 0.1
2 5 0.2
5 3 5
3 4 -9.5
4 5 2.3
1 5 1
```

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 em 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 de um vértice qualquer para todos os outros vértices do grafo, assim como um caminho que possui esta distância (árvore geradora induzida pela busca). Se o grafo possuir pesos não negativos, o algoritmo de Dijkstra deve ser utilizado (caso contrário informar que a biblioteca ainda não implementa caminhos mínimos com pesos negativos).

   Você deve implementar o algoritmo de Dijkstra de duas formas: (1) utilizando um vetor para armazenar as estimativas de distâncias para cada vértice; (2) utilizando um heap para armazenar as estimativas de distâncias para cada vértice. No caso do heap, você pode escolher implementar seu próprio heap ou utilizar uma biblioteca que possua um heap que possa ter as chaves dos seus elementos modificados (atenção com este aspecto).

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 10 e os vértices 20, 30, 40, 50, 60. Apresente os resultados em uma tabela.

2. Determine o tempo médio para calcular a distância entre um vértice e todos os outros vértices do grafo para cada uma das implementações do algoritmo de Dijkstra. Escolha k vértices iniciais (ex. $k = 100$) de forma aleatória, calcule o tempo total de execução, e faça a média amostral. Apresente os resultados em uma tabela.

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 no grafo.