Centro Federal de Educação Tecnológica de Minas Gerais - Departamento de Computação

Laboratório de Algoritmos e Estruturas de Dados II (2ECOM.025)

Lista de Exercícios 10

Professores: Amadeu Almeida e Thiago Rodrigues **Data de entrega:** 18 de Novembro de 2019

Instruções: leia com atenção **todas** as orientações abaixo antes de começar a lista.

- Os exercícios devem ser feitos individualmente ou em dupla e valem 6 pontos.
- Submeta no SIGAA um arquivo compactado chamado **Lista10.zip**. Ele deverá conter:
  - As classes que implementam as soluções das tarefas.
  - Um relatório em **PDF** que inclui os resultados dos experimentos, as tabelas, os grafos e as respostas das questões do exercício 3.
- Caso a tarefa seja feita em dupla, apenas um aluno precisa submetê-la.
- O prazo de entrega é até às 23:59 do dia 18 de Novembro.
- Códigos fonte em PDF e submissões em atraso serão desconsiderados.
- O enunciado desta lista possui 3 exercícios e 3 páginas.

Exercício 1: classe Grafo

Esta tarefa objetiva implementar uma classe que manipula um grafo que pode ser orientado ou não e que é representado por uma matriz de adjacências. Um exemplo de grafo direcionado que possui 4 vértices e 6 arestas é apresentado na figura 1 e a matriz de adjacências deste grafo é exibida na tabela 1.

Grafo:

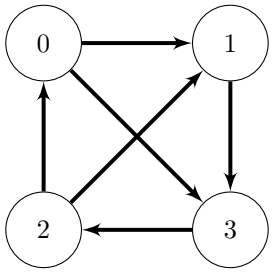

Figura 1: grafo com 4 vértices e 6 arestas.

Matriz de adjacência:

Vértice	0	1	2	3
0	0	5	0	3
1	0	0	0	3
2	6	6	0	0
3	0	0	1	0

Tabela 1: matriz de adjacências do grafo mostrado na figura 1

1. Abra o projeto **ListasDeGrafos** e o arquivo Grafo.java.
2. Adicione o seguinte método na classe **Grafo**:
   (a) public void setPeso(int vertice1, int vertice2, int peso) - altera o peso de uma aresta do grafo.

Exercício 2: classe AlgoritmosEmGrafos

O objetivo deste exercício é acrescentar o algoritmo de Ford-Fulkeson para a obtenção do fluxo máximo em redes à biblioteca de procedimentos em grafos que vem sendo construída na classe **AlgoritmosEmGrafos**.

1. Acesse o arquivo **AlgoritmosEmGrafos.java**.
2. Insira os seguintes atributos na classe **AlgoritmosEmGrafos**:
   - **private final** $ArrayList < ArrayList < Integer >>$ **caminhosDeAumentoFR** - armazena todos os caminhos de aumento que foram encontrados durante a execução do algoritmo de Ford-Fulkeson.
   - **private final** $ArrayList < Integer >$ **capacidadeResidual** - guarda o valor da aresta de menor peso em cada caminho de aumento.
3. Elabore os métodos a seguir:
   - **public AlgoritmosEmGrafos(int vertices)** - o método construtor deve ser atualizado com as inicializações dos novos atributos da classe.
   - **public int iniciaFluxoMaximoEmRedes(int verticeInicial, int verticeFinal)** - inicia o algoritmo que calcula o fluxo máximo entre dois vértices de uma rede.
   - **private int fluxoMaximoEmRedes(int verticeInicial, int verticeFinal)** - implementa o algoritmo de Ford-Fulkeson e retorna o fluxo máximo entre um vértice inicial e outro final.
   - **public** $ArrayList < ArrayList < Integer >>$ **getCaminhosFluxoRedes()**
   - **public** $ArrayList < Integer >$ **getArestaMenorPeso()**
4. Explique o objetivo de todos os métodos implementados no item 3 por meio de comentários no próprio código.

Exercício 3: experimentos computacionais

Este exercício visa testar o funcionamento do algoritmo de Ford-Fulkeson em diferentes tipos de grafos orientados e abordar alguns aspectos relacionados ao problema de fluxo máximo em redes e ao procedimento implementado.

1. Antes de iniciar os experimentos, faça o download dos dois grafos que estão anexados junto a esta lista no SIGAA. A primeira linha de cada arquivo possui, respectivamente, a quantidade de vértices e arestas dos grafos, enquanto as demais contém uma das arestas do grafo. O nó de saída, o de entrada e o peso da aresta são retratados, respectivamente, no primeiro, no segundo e no terceiro número de cada linha.

   Por exemplo, o arquivo que caracteriza o grafo mostrado na primeira página deste documento é organizado da seguinte forma:

```
4 6
0 1 5
0 3 3
1 3 3
2 1 6
2 2 6
3 2 1
```

2. Experimento 1:
   (a) Execute o algoritmo de Ford-Fulkeson para calcular o fluxo máximo entre os vértices 0 e $V-1$ de cada um dos dois grafos, onde V é a quantidade de vértices da rede.
   (b) Para cada grafo, retorne:
      i. O valor do fluxo máximo entre os nós 0 e $V-1$.
      ii. Os vértices que estão em todos os caminhos de aumento entre 0 e $V-1$.
      iii. A capacidade residual de cada um dos caminhos de aumento.
3. Experimento 2:
   (a) Rode o procedimento de Ford-Fulkeson em cada um dos dois grafos para computar o fluxo máximo entre dois vértices distintos a sua escolha.
   (b) Para cada grafo, devolva:
      i. O custo do fluxo máximo entre estes nós.
      ii. Os nós que estão nos caminhos de aumento entre os dois vértices.
      iii. A capacidade residual de todos os caminhos de aumento.
4. Crie duas tabelas que retratem os resultados do experimento 1 para os dois grafos. Cada tabela possui duas colunas contendo respectivamente:
   (a) Todos os vértices pertencentes a cada caminho de aumento.
   (b) A capacidade residual de cada caminho.

   Modelo de Tabela:

Caminho de Aumento	Capacidade Residual

   A tabela abaixo exibe os caminhos de aumento entre os vértices 0 e 3 do grafo apresentado na página 1 e as capacidades residuais de cada um destes caminhos.

Caminho de Aumento	Capacidade Residual
0 1 3	3
0 3	3

5. Desenhe quatro **grafos**:
   (a) Dois que apresentem os caminhos de aumento obtidos durante os experimentos 1 e 2 para o primeiro grafo. Estes grafos devem conter:
      i. Os vértices do grafo original que estão em pelo menos um caminho.
      ii. Todos os caminhos de aumento e suas capacidades residuais.
   (b) Dois que exibam, para o segundo grafo, os caminhos de aumento encontrados no decorrer dos experimentos 1 e 2. Devem estar presentes nestes grafos:
      i. Todos os nós do grafo original que aparecem em ao menos um caminho.
      ii. Os caminhos de aumento com suas respectivas capacidades residuais.
6. Responda as seguintes questões:
   (a) Por que nem todos os vértices e arestas dos grafos originais estão presentes nas redes ilustradas no item anterior?
   (b) Os grafos desenhados no item anterior são árvores? Por quê?
   (c) Para um mesmo grafo, o valor do fluxo máximo encontrado nos experimentos 1 e 2 é igual? Por quê?
   (d) Por que nenhum caminho de aumento entre dois vértices pode ser considerado um caminho mínimo?
   (e) Qual é relação entre a complexidade do algoritmo de Ford-Fulkerson e a quantidade de caminhos de aumento?