

## Entrega 2 - Relatório de Avaliação

### Grupo 2:

- Ionésio Lima da Costa Júnior (115110120)
- José Roberto do Nascimento Júnior (116110170)
- Nicácio Oliveira de Sousa (115111897)

### Descrição

Relatório de avaliação da atividade prática 1 do grupo 11 para a disciplina de Aplicações de Teoria dos Grafos.

### Método de avaliação

A avaliação será o resultado da média aritmética das notas dos quatro critérios adotados. São eles: **Avaliação, Legibilidade, Documentação e Testes**. Em cada um destes critérios (*exceto Testes*) existem observações que classificamos de acordo com a gravidade do erro ou acerto. Estas classificações variam entre **excelente**, **bom**, **médio** e **ruim** e serão decisivas para avaliar a nota final do grupo. Por exemplo, observações que foram classificadas como **excelente** somarão pontos à nota final e, de forma análoga, observações que foram classificadas como **ruim** representam menos pontos na nota final.

### Avaliação

#### 1. Design:

- A biblioteca foi pensada de uma forma bem interessante e que provê um uso genérico para a manipulação dos grafos do tipo **Graph**, que é a principal classe da biblioteca e define a estrutura de um grafo. É interessante citar também **GraphManager**, que é uma classe com métodos estáticos que realiza operações sobre grafos que são passados como parâmetros e **GraphReader**, responsável pela leitura de um grafo através de um arquivo. **(excelente)**
- Não existem contratos para classes importantes como **Graph**. **(médio)**
- A distribuição das classes nos pacotes não faz sentido. *O pacote de testes está contido no pacote de implementação.* **(médio)**

#### 2. Implementação:

- A implementação dos métodos **não podem** tratar as exceções que serão lançadas para quem está instanciando os objetos. **(ruim)**
  - Em **GraphReader.readGraph**, existe um *try...catch* que captura a exceção de um arquivo inválido. Não faz sentido, quem deve resolver a exceção é quem chama o método **readGraph**.
- A utilização de um *Set* de vértices, uma *matriz* e um *Map* para representar as adjacências **adicionam redundância em relação à manipulação da estrutura**. **(ruim)**

#### 3. Legibilidade:

- A maioria dos métodos contém documentação em relação à descrição, parâmetros de entrada e retorno. **(excelente)**

- b. As variáveis definidas nos métodos estão com nomes legíveis e que fazem sentido no escopo. **(excelente)**

#### 4. Testes

- a. Não funciona para vértices com valor 0;
- b. Não existe representação de classe (toString) para as estruturas e a classe Graph;
- c. Não funciona para grafos com valores de nó negativos;
- d. GetVertices deveria retornar null no caso de não possuir vértices;
- e. Em SetWeight o método não trata possíveis valores null na entrada;
- f. GraphRepresentation na representação em matriz não funciona para grafos com peso;
- g. DFS não trata para vértices de entrada com valor null;
- h. É estranho lançar e capturar os erros em um mesmo momento com try/catch aninhados no mesmo local.
- i. shortestPath funcionou bem em grafos sem peso. Entretanto, apresentou estado de loop quando submetido a testes com grafos com pesos (sendo eles somente positivos ou negativos e positivos).
- j. isConnected falhou ao tentar identificar no teste em um grafo que não está completamente conectado.
- k. mst funcionou no teste em grafos sem peso. Entretanto não completou a árvore mínima quando submetido a um grafo com peso.

#### 5. Nota final

- a. **Design:** 8.5
- b. **Implementação:** 7.0
- c. **Legibilidade:** 10.0
- d. **Testes:** 7.0
- e. **Final:** 8.1

#### Cobertura

1. **Porcentagem:** 88,5%
2. **Print:**

Element	Coverage	Covered Instruct	Missed Instruct
Lab1-GRUPO-11	60,8 %	2.054	1.324
Lab1ATG/src	62,1 %	1.568	957
graphLibrary.test.graphLibrary	0,0 %	0	753
graphLibrary	88,5 %	1.568	204
Testes-Grupo-2	57,0 %	486	367

#### Possíveis melhorias

1. Pensando em expansão, podemos fazer uma comparação entre **GraphManager** e a biblioteca **Math** de Java. Ou seja, adicionar operações entre grafos adicionaria funcionalidades bastante úteis à biblioteca de manipulação de grafos do grupo 11.
2. Simplificar a implementação utilizando apenas um **Set** para representar *arestas*, que por sua vez já possuem os respectivos *vértices*.