

## **RELATÓRIO - ATIVIDADE 01 - GRAFOS**

Grupo:

Daniela Heckler Mello - 17200438

Eduardo Dias Gutterres - 17200439

### **Exercício 01 - Representação**

Para a realização desse exercício foram criadas as classes Main, Grafos, Vértice e Aresta, cada uma das quais com seus atributos e métodos. Para melhor organizar as estruturas também foi utilizado ArrayList. Toda a criação do grafo, lido pelo arquivo de entrada, foi configurado no método construtor, enquanto os outros métodos de busca de informações apenas retornam o dado capturado.

### **Exercício 02 - Busca**

Para realizar a busca, foi utilizado um laço for para setar as configurações iniciais do grafo. Então, foi utilizada uma fila de Vértice para organizar a ordem de execução das visitas para cada vértice do grafo. Para isso, o algoritmo recebe o arquivo Grafo como parâmetro do método, e busca a lista de vértices do grafo recebido.

### **Exercício 03 - Ciclo Euleriano**

Na realização do algoritmo do Ciclo Euleriano, foi utilizado uma estrutura de HashMap, em que relacionamos a classe Aresta e Boolean, setando false para todas as arestas. Seleccionamos um vértice do conjunto de vértices do grafo e então, criamos um Pair de Boolean com a lista de Vértices. Caso o boolean seja falso, retorna a estrutura Pair com valor boolean falso e nulo. Do contrário, utiliza um laço de repetição for para encontrar outros booleans. Se encontrar um falso, retorna uma mensagem de erro e outro par com boolean valor falso e nulo. Se não encontrar um boolean falso, segue até retornar true e o ciclo Euleriano.

### **Exercício 04 - Algoritmo de Bellman-Ford ou de Dijkstra**

No algoritmo de BellmanFord, iniciamos por realizar um for para a setagem das configurações iniciais, com todos os pesos para infinito, antecessores como nulo e visitado como false. Então, utilizamos outros dois laços for para percorrer toda a lista de vértices e arestas, setando o caminho mínimo para cada um dos casos. Após isso, verificamos se o peso foi setado de maneira correta, retornando um erro caso esteja incorreta. Por fim, utilizamos um for percorrendo todos os vértices do grafo, para realizar a impressão do resultado na tela.

### **Exercício 05 - Algoritmo de Floyd-Warshall**

Para a implementação do algoritmo foi realizada a criação de uma matriz de pesos, a partir dos dados retirados do Grafo gerado pela leitura do arquivo. A partir daí, foram

utilizados três laços de repetição *for* para comparação e setagem dos pesos de cada par de vértices. Apenas uma matriz foi utilizada para isso, substituindo os dados sempre que encontrar um caminho menor.