UNIVERSIDADE FEDERAL DE SANTA CATARINA

INE5413 - GRAFOS

RELATÓRIO DA ATIVIDADE 1

BRENDA SILVA MACHADO

2023.1

1. Código Fonte

A linguagem de programação escolhida para a realização desta atividade foi Python, voltada para a Orientação Objetos. Não foi necessário o uso de nenhuma biblioteca. O código fonte da atividade está disponível no GitHub, no link: https://github.com/Brenda-Machado/grafos

2. Representação

A primeira questão da atividade pedia a implementação de um tipo estruturado de dados ou uma classe que representasse um grafo. Foi escolhido a representação por uma classe, a qual não recebe argumentos. Através da sua instância, pode-se chamar o método ler para receber um arquivo como argumento e, caso esteja no formato certo, passar os conjuntos de valores identificados para suas respectivas variáveis na classe Grafo.

A escolha da estrutura de classes tem como objetivo se adaptar à estrutura de Orientação a Objeto que o Python possui, permitindo maior agilidade, uma vez que já estou familiarizada com isso, e um minimalismo na implementação, permitindo evitar redundância no código.

Os vértices e a função foram implementados na forma de dicionário, uma vez que a presença de uma chave facilita a identificação dos elementos. Já as arestas foram implementadas na forma de lista, uma vez que receberia dois elementos, os vértices que liga, na forma de tupla, ex. (1,2).

3. Buscas

A segunda questão pede um algoritmo que realiza busca em largura a partir de um grafo de entrada. A forma de estruturação usada foi uma classe, em que o método busca faz a busca em largura propriamente dita. As variáveis distância e nodo são dicionários, enquanto fila e visitados são listas. O dicionário por causa da presença de chaves identificadoras, permitindo verificar a quem pertence aquela distância/nodo, simulando um atributo. Já as listas por causa da simplicidade e por conseguirem simular as filas do algoritmo original, com ligeira alteração nos métodos.

4. Ciclo Euleriano

Para a implementação do ciclo euleriano, a estrutura de classes também foi seguida. Para a implementação do ciclo, o qual teria que ser retornado numa certa

ordem, foi escolhida a lista, uma vez que ela é capaz de manter a ordem dos elementos tal qual são inseridos. Além disso, as listas em python são simples e eu domino seu uso, o que facilita a implementação. Já o visitados, também lista, só armazena os vértices que já foram visitados, a escolha teve o acréscimo da facilidade de remover um elemento em específico da lista.

5. Bellman-Ford

O algoritmo de Bellman-Ford também seguiu a estrutura de classes, porém as suas variáveis são todas baseadas em dicionários, ao contrário das outras questões. O motivo para isso é que há vários vértices, nodos e caminhos sendo analisados, e usar uma estrutura baseada em chaves facilita a verificação da existência dos valores e quais valores estão contidos. Também, é uma estrutura que é fácil de ser varrida e seus valores mudados, o que ocorre com frequência ao longo do algoritmo.

6. Floyd-Warshall

Com Floyd-Warshall, as listas foram utilizadas para criar as matrizes do algoritmo. As listas são uma ferramenta bem útil para a criação de matrizes em python, permitindo uma forma minimalista de manipulação e varredura, além de facilidade de implementação.