

INE5413 - Atividade 1

Gabriel da Silva Cardoso e Hans Buss Heidemann

Maio 2022

1 Código-fonte

O código utilizado está disponível no Github em <https://github.com/Cardoz-0/Graph-Theory-Study>

2 Representação

A linguagem escolhida para este trabalho foi Python pela sua agilidade de implementação, que é uma linguagem orientada a objeto. Para representar o problema foram utilizadas três classes:

- Graph: Contém apenas os vértices e informação sobre ser dirigido ou não. A classe grafo que acessará os vértices para adicionar as arestas. Vale ressaltar que os vertices são armazenados em lista, logo podem ser representados na forma de um index.
- Vértice: Os vértices contém informações sobre o seu rótulo e suas arestas, também em forma de lista
- Edge: As arestas são objetos que apontam para o vertice de entrada e o de saída, junto com o seu peso.

Foram utilizadas algumas funções auxiliares para mostrar o resultado por meio de texto no console, como *vertice-to-index*, que transforma objetos de vertice da lista em posições, facilitando a visualização.

3 Breadth-First Search

Para implementar o algoritmo de busca em largura foi utilizada uma fila e uma lista. A lista mantém informações sobre os vertices que já foram visitados pelo algoritmo e a fila adiciona os vertices que devem ser visitados (caso já não tenham sido) nas próximas iterações do algoritmo. Por ser um algoritmo sem peso ele se torna bastante simples, logo, não precisa de muitas estruturas de dados.

4 Hierholzer Algorithm

Para detectar um ciclo euleriano utiliza-se apenas uma lista de nodos visitados, como no exercicio anterior e uma lista auxiliar para saber os vertices que pertecem ao ciclo.

5 Dijkstra

Para o algoritmo de Dijkstra experimentou-se usar dicionarios para poder acessar as informações sobre o nodo diretamente com o seu objeto, sem precisar fazer uma transformação para index. Esses dicionários mantêm informações sobre distancia e melhor caminho cara cada vertice do grafo. Alem disso, foi utilizada novamente uma lista de nodos visitados. Como resultado do uso de dicionario, foi precisado de um código extra para conseguir transformar a saída em texto no terminal.

6 Floyd-Warshall

Floyd-Warshall utiliza como principal estrutura de dados uma lista uv para manter informações sobre os pares de caminhos. No caso de nossa implementação, ao invés de uma matriz é utilizado um dicionario com todos os pares de entradas, por simplicidade na implementação. Por floyd warshall ser uma implementação mais simples, não são necessárias outras estruturas de dados