

Atividade 1 de Grafos
Universidade Federal de Santa Catarina
Ciências da Computação
Outubro de 2022

Eric Fernandes Evaristo - 21250089
Matheus Aparicio da Silva - 20100538
Otávio Wada - 20100540

1. Representação:

O trabalho foi desenvolvido na linguagem Python 3.10 e foram utilizadas classes para representar os grafos e os vértices. A classe grafo (*Graph*) tem dois atributos, uma lista de vértices e uma matriz que representa as arestas. A classe vértice (*Vertex*) tem três atributos, o rótulo (*label*), grau (*degree*) e a lista de vizinhos (*neighbors*).

2. Buscas:

Para esta questão foi utilizada a busca em largura. Devido à propriedade de propagação deste algoritmo, ele se torna o melhor candidato para a classificação dos vértices em níveis. Algumas modificações foram feitas no algoritmo, como o uso do dicionário no lugar de listas para os vértices antecessores e as distâncias. Isso torna a implementação do programa final mais simples, pois não exige a reordenação dos dados.

3. Ciclo Euleriano:

Para a resolução deste problema foi usado o algoritmo de Hierholzer, responsável por identificar o ciclo Euleriano em tempo $O(|E|)$. O algoritmo responsável por buscar o subciclo está contido dentro da função `hierholzer()`, não tendo sido declarada externamente.

4. Algoritmo de Bellman-Ford ou de Dijkstra:

Neste exercício o algoritmo escolhido foi o de Dijkstra. Este algoritmo é um pouco mais eficiente que o Bellman-Ford, por isso ele se torna um bom candidato na resolução do problema, a única desvantagem é a falta de suporte para grafos com pesos negativos. O programa usa a lista de antecessores para montar o caminho juntamente com a lista de distâncias para exibição do custo de cada caminho.

5. Algoritmo de Floyd-Warshall:

O algoritmo de Floyd-Warshall implementa a identificação dos pesos dos caminhos mínimos presentes no grafo. Algumas de suas desvantagens são que não identifica o caminho em si, e o maior consumo de memória devido à alocação de uma matriz por vértice presente.