Grafos

Leonardo Brito — Lívia Corazza — Rita Magalhães

1 Representação

A classe `Grafos` lê um arquivo de entrada contendo informações sobre um grafo e cria uma lista de adjacência para representar o mesmo.

As estruturas utilizadas nesta classe são: Dicionários para armazenar a lista de adjacência do grafo, onde a chave é o rótulo do vértice e o valor é uma lista de tuplas que representa as arestas com o rotulo do vizinho e o respetivo peso; Listas para armazenar os rótulos dos vértices e os vizinhos; e por fim, variáveis inteiras para armazenar o número de vértices e o número de arestas do grafo. Foi criada também uma matriz de adjacência que é outra forma de representar o grafo e será utilizada no algoritmo de Floyd-Warshal.

2 Buscas

O algoritmo de busca em largura (BFS - Breadth-First Search) inicia a busca a partir de um vértice inicial e explora todos os vértices adjacentes de nível mais próximo antes de se mover para os vértices de nível seguinte. Os vértices vão sendo armazenados em uma fila. A complexidade de tempo da BFS é O($|V| + |E|$), onde V é o número de vértices e E é o número de arestas no grafo.

3 Ciclo Euleriano

Foi utilizado o algoritmo de Hierholzer para resolver esse problema, que é capaz de identificar o ciclo Euleriano em tempo O($|E|$). A função `ciclo_euleriano` contém o algoritmo responsável por montar o ciclo principal utilizando duas outras funções que descobrem e montam subciclos, retornando o resultado em formato booleano.

4 Algoritmo de Dijkstra

O algoritmo utilizado foi o de Dijkstra, o qual é mais rápido em relação a Bellman-Ford. O algoritmo utiliza a lista de distâncias e a lista de antecessores para exibir os custos de cada caminho e encontrar o menor caminho. Uma desvantagem é o fato do algoritmo não funcionar quando há aresta de peso negativo.

5 Algoritmo de Floyd-Warshall

O algoritmo descobre os caminhos mais curtos entre todos pares de vértices. Para tal, o loop exterior faz iterações sobre todos os vértices intermédios possíveis e o loop interior faz iterações sobre todos os pares de vértices para actualizar a distância mais curta dado o vértice intermédio. A complexidade do algoritmo é O($|V|^3$) porque faz iterações $|V|$ no loop exterior, e para cada iteração, actualiza todas as entradas da matriz, o que requer V^2 operações.