

Relatório Grafos

Bruno Carvalho Caxias

Estratégia utilizada: Lista de Arestas

A estratégia para implementação do trabalho de grafos foi feita por lista de arestas. Nesse modelo a classe grafo guarda 2 listas, uma de vértices contendo todos os vértices que compõem o grafo e uma lista de aresta guardando todos as arestas no grafo (levando no sentido de cidades cada vértice seria uma cidade e cada aresta seria um caminho).

A lista de arestas é desconectada da lista de vértices, isso ocorre para permitir a existência de vértices desconexos. Além disso, essa forma de implementação facilita a representação de laços e arestas paralelas.

A classe de Aresta guarda as informações de onde é sua origem e qual é seu destino, além de guardar também o peso da mesma (no sentido de cidade seria sua distância), fora isso em questão de métodos ela só possui seus getters, setters e constructor.

A classe vértice também é incrivelmente simples guardando somente o valor que será atribuído aos vértices e possuindo como métodos os getters, setters e constructor.

Em relação aos métodos utilizados para implementação do grafo, ambos métodos de inserir vértice e arestas somente adicionam os mesmos as listas, para se obter as arestas de um vértice é percorrida a lista de arestas fazendo comparações para saber quais arestas possuem o vértice de origem similar e os métodos de busca em largura percorrem os vértices utilizando 2 listas criadas no próprio método, uma de fila para saber quais vértices devem ser visitados em sequencia e outra para saber quais vértices já foram visitados.

Métodos de busca de largura possuem complexidade de $O(n^2)$ devido a chamada do método de busca de destinos e por isso são uma das partes mais demoradas do código.