

Marcos Cesar Ribeiro de Camargo - N° USP: 9278045

Implementação do TAD lista de adjacências e algoritmo de Floyd Warshall

São Carlos – SP

07 de Junho de 2016

Sumário

1	Introdução	3
2	Descrição	3
2.1	Estrutura de dados	3
2.2	Estrutura Lógica	3
	REFERÊNCIAS	5

1 Introdução

O trabalho visa resolver o problema do caminho mínimo, que é o caminho de menor custo, entre dois vértices v e w . Para modelar esse problema foi implementado uma estrutura de dados dinâmica para grafos não direcionados na forma de lista de adjacências e para encontrar o caminho de menor custo entre dois vértices quaisquer foi utilizado o algoritmo de Floyd Warshall.

2 Descrição

2.1 Estrutura de dados

Para a implementação do projeto foram utilizadas algumas estruturas de dados: Uma lista genérica, que foi utilizada para a lista de vértices, arestas e adjacências; Um vértice, que armazena o seu valor e sua lista de adjacências; Uma aresta, que armazena um valor real, referências para seus vértices finais e as referências para os seus respectivos nós da lista de adjacências; E um adjacente, que armazena a referência para o nó na lista de arestas. Todas as estruturas acima armazenam uma referência para o seu nó da sua respectiva lista genérica, para que a remoção possa feita em tempo constante; E por ultimo, um mapa implementado em lista que armazena uma tripla, formada por (identificador, indice da matriz do algoritmo de floyd warshall, referência).

2.2 Estrutura Lógica

O projeto funciona a partir de um menu principal, que lê uma string de 2 caracteres da stdin, e A decodifica para saber qual a operação desejada. Neste projeto foram utilizadas 11 operações básicas:

- a) FM - termina a execução do programa, saindo do menu e destruindo as estruturas criadas.
- b) CV - cria um novo vértice, com um valor passado pela stdin. A criação do vértice é feita a partir da função de inserir vértice do TAD grafo. Após a inserção do vértice no grafo, uma referência para o vértice é adicionada a um mapa, facilitando a identificação pelo usuário.
- c) DV - deleta um vértice pelo identificador lido na stdin. A remoção do vértice é feita a partir da função de remover vértice do TAD grafo, que remove o vértice e todas aresta que terminam nele. A referência para o vértice é encontrado no mapa pelo identificador fornecido via stdin.
- d) CA - cria nova aresta, com os identificadores dos vértices finais e o elemento armazenado lidos da stdin. A inserção da aresta é feita a partir da função de

inserir aresta do TAD grafo, e após a inserção da aresta no grafo, uma referência da aresta é adicionada ao mapa para identificação posterior.

- e) DA - deleta aresta pelo identificador lido da stdin. A remoção da aresta é feita a partir da função de remover aresta do TAD grafo, a referência para a aresta se encontra no mapa, sendo identificada pelo valor lido da stdin.
- f) TV - troca o valor armazenado em um vértice por um novo valor, tanto o identificador quanto o novo valor são lidos da stdin. Para a troca é utilizado a função de substituição de valor de um vértice do TAD grafo.
- g) TA - troca o valor armazenado em uma aresta por um novo valor, tanto o identificador quanto o novo valor são lidos da stdin. Para a troca é utilizado a função de substituição de valor de uma aresta do TAD grafo.
- h) FW - executa o algoritmo de Floyd Warshall, obtendo todas as distâncias mínimas entre os vértices e seus caminhos. O algoritmo utiliza a operação de obter os vértices finais de uma aresta do TAD grafo, e também utiliza duas funções auxiliares que retornam referência para a lista de vértices e arestas, para que ele possa percorrer essa lista. As matrizes de distância e parentesco são inicializadas e desalocadas por funções implementadas separadamente do TAD grafo.
- i) IG - imprime o grafo. Utiliza a operação de obter os vértices finais de uma aresta e a operação de obter os valores armazenados em um vértice/aresta do TAD grafo, e também utiliza duas funções auxiliares que retornam referência para a lista de vértices e arestas, para que ele possa percorrer essa lista. Os valores de identificação dos vértices e arestas estão armazenados no mapa.
- j) ID - imprime a distância mínima entre dois vértices, os quais os identificadores são lidos da stdin. Utiliza a matriz de distância obtida pelo algoritmo de Floyd Warshall. Os valores de identificação dos vértices estão armazenados no mapa.
- k) IC - imprime o caminho mínimo entre dois vértices, os quais os identificadores são lidos da stdin.

O algoritmo de Floyd Warshall foi implementado baseado no algoritmo visto no [Wikipedia \(2016\)](#), e o TAD Lista de Adjacências foi implementado de acordo com os slides passados pelo [Prof.Campello \(2016\)](#).

Referências

PROF.CAMPELLO. *TAD*. 2016. [Online; accessed 07-June-2016]. Disponível em: http://disciplinas.stoa.usp.br/pluginfile.php/728932/mod_resource/content/2/Grafos_II_Representa%C3%A7%C3%B5es.pdf. Citado na página 4.

WIKIPEDIA. *Floyd–Warshall Algorithm* — *Wikipedia, The Free Encyclopedia*. 2016. [Online; accessed 07-June-2016]. Disponível em: https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm. Citado na página 4.