Relatório do Trabalho sobre Caminhos Mínimos

Universidade de São (USP)
Instituto de Ciências Matemáticas e Computação (ICMC)
São Carlos, SP
Junho/2013

Elias Italiano Rodrigues (7987251)
Modelagem Computacional em Grafos (SCC0216)

Resumo

Este relatório descreve a implementação de um Tipo Abstrato de Dados (TAD) para a representação e implementação de um Grafo (direcionado ou não) em linguagem C. Descreve também a utilização deste TAD para resolver o problema descrito no trabalho da disciplina de Modelagem Computacional em Grafos que envolve buscar por caminhos mínimos aplicando-se dados de heurística.

Implementação do TAD

Foram definidas três estruturas de dados, `graph_t`, `vertex_t` e `edge_t`, para representar, respectivamente, um grafo, um vértice e uma aresta. O TAD de grafo foi implementado em linguagem C da seguinte maneira:

- Um grafo contém um vetor de vértices (ponteiros)
- Cada vértice contém um vetor de arestas (ponteiros)

Ao fazer uma ligação entre dois vértices *u* e *v* em um grafo não-direcionado, uma aresta é criada de *u* para *v* e outra de *v* para *u*. Além das operações básicas, o TAD também implementa operações de adicionar vértices e criar arestas a partir de um arquivo de texto que facilitaram, em específico, a importação dos dados do problema descrito no trabalho. Segue a lista de operações do grafo:

- Inicializar um grafo
- Finalizar um grafo
- Criar N vértices
- Adicionar um vértice
- Adicionar vértices a partir de um arquivo de texto
- Alterar informações de um vértice
- Remover um vértice
- Obter o ponteiro de um vértice
- Conferir se um vértice existe
- Obter a quantidade atual de vértices
- Criar uma aresta
- Criar arestas a partir de um arquivo de texto
- Remover uma aresta
- Obter o ponteiro de uma aresta
- Conferir se uma aresta existe

- Alterar o peso de uma aresta
- Imprimir o grafo no formato de uma matriz
- Executar busca em largura (BFS)
- Executar busca em profundidade (DFS)
- Executar busca por caminhos mínimos (Dijkstra)
- Imprimir o caminho entre dois vértices

Utilização do TAD para resolver o problema

Dado um grafo em que os vértices representam as cidade capitais do Brasil e as arestas representam as distâncias entre elas pela rodovia, o problema consiste em encontrar caminhos mínimos entre uma cidade e outra. Para resolvê-lo, usou-se o algoritmo de Dijkstra e mais um conjuto de dados de heurística sobre a distância em linha reta entre as cidades.

O TAD foi implementado de tal maneira que cabe a resolução deste problema e ainda mantém-se genérico para a solução de outros problemas quaisquer que envolvam o algoritmo de Dijkstra padrão. A função que executa o algoritmo de Dijkstra tem a seguinte assinatura:

```
int graph_dijkstra(graph_t *g, int s, double (*func)(int));
```

Ela pode receber como último parâmetro a referência de um função que retorna um dado (peso) de heurística. E passando-se uma referência nula `NULL`, a função segue com o algoritmo padrão de Dijkstra visto nas aulas.

Então, a utilização do TAD consistiu em:

- Criar e inicializar o grafo
- Adicionar os vértices a partir de um arquivo de texto (Tabela 1 da descrição do trabalho).
- Criar as arestas a partir de um arquivo de texto (Tabela 2 da descrição do trabalho)
- Executar o algoritmo de Dijkstra passando um função que consulta os dados de heurística (Tabela 3 da descrição do trabalho) e retorna a distância.
- Imprimir na tela os resultados para análise.
- Finalizar o grafo

Foi executado também a busca usando o algoritmo de Dijkstra padrão para poder comparar contra o algoritmo que usa os dados de heurística.

Análise dos resultados obtidos

Executando o programa final para todas as possibilidades possíveis (conferir o arquivo de tabela que acompanha este com todos os casos) e analisando os resultados, é possível perceber que a execução do algoritmo usando os dados de heurística consegue obter a resposta com um número de operações menor ou igual ao do algoritmo padrão. É possível observar também, que em alguns casos particulares, ocorre de a distância do caminho obtido utilizando-se os dados de heurística ser ligeiramente maior (porém sempre com o número de operações menor). Possivelmente, esse efeito pode ter sido provocado por alguma imprecisão nos dados de heurística (distância em linha reta entre as cidades).