

Rudini Menezes Sampaio
Aluno do Instituto Tecnológico de Aeronáutica (ITA)
Bolsa PIBIC/CNPq
Orientador: Horácio Hideki Yanasse
Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)
Avenida dos Astronautas, 1758 - Caixa Postal 515

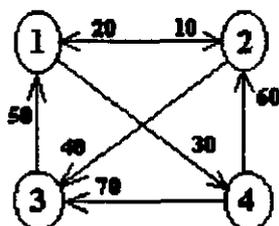
Problemas urbanos práticos tais como entrega de correspondência, limpeza de rua, coleta de lixo, serviços de ônibus para melhor atendimento à população, reparos de cabines telefônicas, remoção de neve e problemas de seleção de lugares para localização de serviços de emergência (pronto socorros, bombeiros,...) ou não emergência (terminais de transporte e etc...) podem ser modelados como problemas em grafos.

A teoria de grafos direcionada para a solução de tais problemas possui várias vantagens entre elas a fácil descrição do problema para parâmetros da teoria e a facilidade com que se pode encontrar soluções aproximadas para problemas matematicamente complexos.

O rápido avanço na tecnologia dos computadores proporcionou uma atenção especial para esta teoria, ocasionando o desenvolvimento de vários algoritmos que solucionem estes problemas. Tais algoritmos envolvem problemas de menor caminho, árvores de custo mínimo, tour euleriano e caminho euleriano, carteiro chinês, caixeiro viajante e outros.

Sendo assim, este trabalho tem por objetivo a implementação de tais algoritmos, criando um ambiente em que se possa manipular grafos e onde seja possível a visualização de tais grafos e dos resultados da aplicação dos algoritmos sobre eles.

Para isto, foi preciso definir uma estrutura de dados para a entrada de grafos, o que foi feito a partir de arquivos textos com um cabeçalho nas primeiras linhas para identificação interna e, nas restantes, cada linha representa um nó informando os nós a quem este se liga seguido do valor do arco correspondente. Para exemplificar, o grafo da figura 1, vem seguido do respectivo arquivo texto que o representa. A forma de desenho do grafo na tela é resolvido durante o desenvolvimento do programa e está bem explicado no relatório final, bem como no "Help" do programa executável, inclusive com as melhorias de tal estrutura.



Esboço do arquivo de representação:

2 10 4 30 0	(Ligações do Nó 1)
1 20 3 40 0	(Ligações do Nó 2)
1 50 0	(Ligações do Nó 3)
2 60 3 70 0	(Ligações do Nó 4)

Figura 1: Grafo seguido do esboço do arquivo que o representa, para exemplificar a estrutura de dados utilizada.

Resolvido o problema da entrada de dados, procurou-se desenvolver um programa com boa ambientação que possibilitasse a visualização dos grafos (entrados segundo a estrutura definida), a manipulação de seus valores e a execução de vários algoritmos. Começou-se então o desenvolvimento do programa "Graph.Pas".

Para tanto, necessitou-se de um conhecimento aprofundado das linguagens Pascal e Assembler, para implementação das rotinas e de tarefas mais complexas tais como gravação da tela em espaços de memória e utilização do mouse.

O programa foi desenvolvido visando uma boa modularização e principalmente a sua robustez (não ocorrência de defeitos, mesmo para entradas com erros), a sua reusabilidade (reutilização de rotinas) e a sua manutenibilidade (condição para introdução de novas rotinas ou modificação das anteriores, sem interferir no funcionamento geral do programa), que são palavras-chave em processos de engenharia de software, quando se busca boa qualidade para o produto final.

Foram necessários muitos testes para verificação do funcionamento correto e completo dos algoritmos, buscando-se todas as possibilidades que levassem a resultados diferentes.

Tendo em vista o tamanho do programa, este foi dividido em sete unidades e um programa principal. Tais unidades podem ser reutilizadas por outros programas. Na execução do programa principal, permite-se carregar um grafo externo, criar um grafo novo, modificar parâmetros do grafo e salvar modificações feitas. Permite-se ainda escolher a forma de visualização do grafo, identificar as principais características do grafo tais como se ele é direcionado, se é conexo, se é fortemente conexo ou se possui tour ou caminho euleriano, e executar os seguintes algoritmos: menor caminho de Dijkstra (entre um dado nó e os restantes), menor caminho de Floyd (entre todos os nós), árvore de custo mínimo de Prim e carteiro chinês de Edmonds.

Um último passo seria o aperfeiçoamento do programa, a implementação de novos algoritmos envolvendo vários problemas, como o de fluxo em grafos e o do caixeiro-viajante, e o desenvolvimento de um programa que utilize dados de imagens de satélite e os transforme para o tipo de entrada de dados que o programa final reconhece.

Bibliografia:

- Aho, A.V.; Hopcroft, J.E.; Ullman, J.D., **The design and analysis of computer algorithms**, Addison-Wesley, 1974.
- Boaventura Netto, P.O., **Grafos: teoria, modelos e algoritmos**, Edgard Blücher, 1996.
- Campello, R.E.; Maculan, N., **Algoritmos e heurísticas**, EDUFF, 1994.
- Duncan, R., **Advanced MSDOS programming**, Microsoft Press, 1986.
- Larson, R.C.; Odoni, A.R., **Urban operations research**, Prentice Hall, 1981.
- Mokarzel, F.C., **Apostila do curso Estrutura de dados**, Computação, ITA, 1990.