

SCC0203 - ALGORITMOS E ESTRUTURAS DE DADOS II

TRABALHO PRÁTICO N. 1

Profa. Debora Medeiros

"O trânsito de Belo Horizonte fica ainda mais complicado quando a sinalização em determinados bairros é insuficiente."

(<http://www.portaldotransito.com.br/noticias/sinalizacao-precaria-aumenta-o-risco-de-acidentes-em-bh.html>).

"Não é difícil ver. Basta ficar olhando por cinco minutos que alguma coisa acontece. O trânsito é muito intenso, vem de todas as direções e o que piora é o fato de que não há sinalização nenhuma, nem vertical, nem horizontal" (afirma o aposentado Paulo Virgílio, 50).

Leia mais: <http://diariodovale.uol.com.br/noticias/0,36088.html#ixzz1GceZFDy9>

Os depoimentos acima ilustram uma triste realidade do trânsito em grandes cidades brasileiras, a falta de sinalização. Como solução para este problema, um pequeno aparelho começou a crescer no mercado de eletrônicos, o GPS (Sistema de Posicionamento Global). Este aparelho recebe o destino como dado de entrada, sendo este informado pelo usuário, então ele toma como base, a localização do motorista e calcula o caminho a ser percorrido. Mas se o motorista quiser optar por não passar por uma determinada via?

1. Descrição do Trabalho

O objetivo deste trabalho é tentar simular de maneira simplificada o funcionamento de um GPS em um grafo onde as arestas representam ruas e avenidas e os vértices representam as esquinas de uma cidade.

1.1 Entrada

O programa a ser desenvolvido deverá ser capaz de receber como dados de entrada:

- a descrição do grafo contendo: número de vértices (n), número de arestas (m) e descrição das arestas com seus respectivos pesos e

- os caminhos requisitados pelo usuário contendo: número de caminhos, descrição de cada caminho (vértice de origem, vértice de destino, número de arestas que não devem pertencer ao caminho solicitado e quais as arestas).

1.2 Processamento

Com base no grafo recebido como entrada, o programa deve calcular os menores caminhos entre todos os pares possíveis de vértices (sequência de vértices e peso total do caminho para cada par), utilizando o algoritmo Dijkstra. Dessa forma, o aplicativo obtém uma base de caminhos e pode responder rapidamente às requisições básicas (onde o usuário não possui objeção por nenhum caminho).

Para cada caminho requisitado, o programa deve verificar se o usuário especificou alguma aresta pela qual não deseja passar. Caso ele não tenha especificado nenhuma ou tenha especificado alguma que não pertença ao caminho original, o programa deve retornar o caminho original, juntamente com seu peso total (soma dos pesos das arestas pertencentes ao caminho). Caso o usuário tenha especificado alguma(s) aresta(s) que pertença(m) ao caminho original, um novo caminho mínimo deve ser calculado, desconsiderando a(s) aresta(s) especificada(s), utilizando o algoritmo Dijkstra.

1.3 Saída

O programa deve retornar:

- uma matriz D , $n \times n$, onde d_{ik} é o peso do menor caminho do vértices i para o vértice k ;
- uma matriz Π , também $n \times n$, onde π_{ik} é o vértice antecessor do vértice k no menor caminho entre do vértice i para o vértice k e
- para cada caminho requisitado, o programa deve informar seu peso total e a sequência de vértices, ou, informar que tal caminho não existe, se for o caso.

2. Padrões de Entrada e Saída

A entrada do programa deve ser lida da entrada padrão (teclado) e a saída deve ser escrita na saída padrão (tela).

A **entrada** deve seguir o seguinte padrão:

$$\begin{array}{l}
n \ m \ r \\
va_1 \ vb_1 \ p_1 \\
va_2 \ vb_2 \ p_2 \\
\dots \\
va_m \ vb_m \ p_m \\
e_1 \ o_1 \ d_1 \ a_{1,1} \ a_{1,2} \ \dots \ a_{e1} \\
\dots \\
e_r \ o_r \ d_r \ a_{r,1} \ a_{r,2} \ \dots \ a_{er}
\end{array}$$

onde:

- n : número de vértices;
- m : número de arestas;
- r : número de caminhos requisitados;
- va_i : vértice onde a aresta i diverge;
- vb_i : vértice onde a aresta i incide;
- p_i : peso da aresta i ;
- e_j : número de arestas indesejáveis do j -ésimo caminho requisitado;
- o_j : vértice origem do j -ésimo caminho requisitado;
- d_j : vértice destino do j -ésimo caminho requisitado e
- $a_{j,l}$: l -ésima aresta indesejável do j -ésimo caminho requisitado (esses valores são ausentes caso $e_j = 0$)

A **saída** deve seguir o seguinte padrão:

$$\begin{array}{l}
d_{1,1} \ \dots \ d_{n,1} \\
\dots \\
d_{n,1} \ \dots \ d_{n,n} \\
\pi_{1,1} \ \dots \ \pi_{n,1} \\
\dots \\
\pi_{n,1} \ \dots \ \pi_{n,n} \\
t_1 \ c_1 \ v_{1,1} \ \dots \ v_{1,c1} \\
\dots \\
t_r \ c_r \ v_{r,1} \ \dots \ v_{r,c1}
\end{array}$$

onde:

- $d_{i,j}$: peso do menor caminho original do vértices i para o vértice j ;
- $\pi_{i,k}$: vértice antecessor do vértice k no menor caminho original do vértice i para o vértice k ($\pi_{i,k} = i$ caso $i = k$);
- t_j : peso total do j -ésimo caminho requisitado (caso o caminho seja inexistente, $t_j = -1$)
- c_j : número de vértices do j -ésimo caminho requisitado (ausente caso o caminho seja inexistente) e
- $v_{j,l}$: l -ésimo vértice do j -ésimo caminho requisitado (ausente caso o caminho seja inexistente)¹.

¹ Os vértices o_j e d_j devem pertencer a essa sequência.

Também devem ser respeitadas as seguintes restrições:

- todos os valores devem ser inteiros;
- os vértices e arestas são numerados sequencialmente começando por 1.
- deve haver somente um espaço em branco entre cada dois números e
- não devem haver linhas vazias.

A seguir é apresentado um exemplo de entrada e saída seguindo o padrão apresentado:

Entrada	Saída
3 4 3	0 7 5
1 2 8	3 0 8
1 3 5	5 2 0
2 1 3	- 3 1
3 2 2	2 - 1
1 1 2 2	2 3 -
1 2 3 3	8 2 1 2
0 1 3	-1
	5 2 1 3

3. Observações importantes

- O endereço de entrega do trabalho será especificado posteriormente.
- O programa deverá respeitar exatamente os padrões de entrada e saída, pois a correção será automática;
- Os códigos serão verificados para se detectar ocorrências de plágio.
- Implemente usando a linguagem C.
- Vazamento de memória, referência a valores de variáveis não inicializadas e outros defeitos serão levados em conta na avaliação;
- Os trabalhos deverão ser feitos em duplas.
- Elabore um relatório que discuta em linhas gerais a estrutura lógica usada para desenvolver o trabalho, sem apresentar código. **Não** entregue o código com o relatório. O relatório deve permitir compreender o trabalho sem a necessidade do código. Entregar o relatório pelo mesmo endereço de entrega do trabalho (a ser especificado posteriormente).
- Trabalhos estão sujeitos a avaliação oral segundo critério do professor e do estagiário PAE.
- Serão anulados aqueles trabalhos nos quais forem detectados quaisquer tipos de cópia ou plágio, não importa a origem.
- A data limite para entrega é 07/05/2010.