SCC216 – Modelagem Computacional em Grafos

Prof.ª Maria Cristina Ferreira de Oliveira

1º sem. 2014

Trabalho 1 – Percursos do Metrô

Prazo de entrega: 31/03/2014 no SSP

Introdução

Pierre é um estudante de intercâmbio que acabou de desembarcar em São Paulo. Como não conhece a cidade, não sabe quais linhas de metrô pegar para chegar ao seu instituto na USP. Faça um programa que, dadas as informações das linhas de metrô da cidade, um ponto de início e um ponto de destino, encontre o caminho mais curto para Pierre. Utilize uma busca em largura para encontrar o caminho mais curto entre os dois vértices do grafo.

Considere que cada linha do metrô liga apenas dois lugares (ver Figura 1).

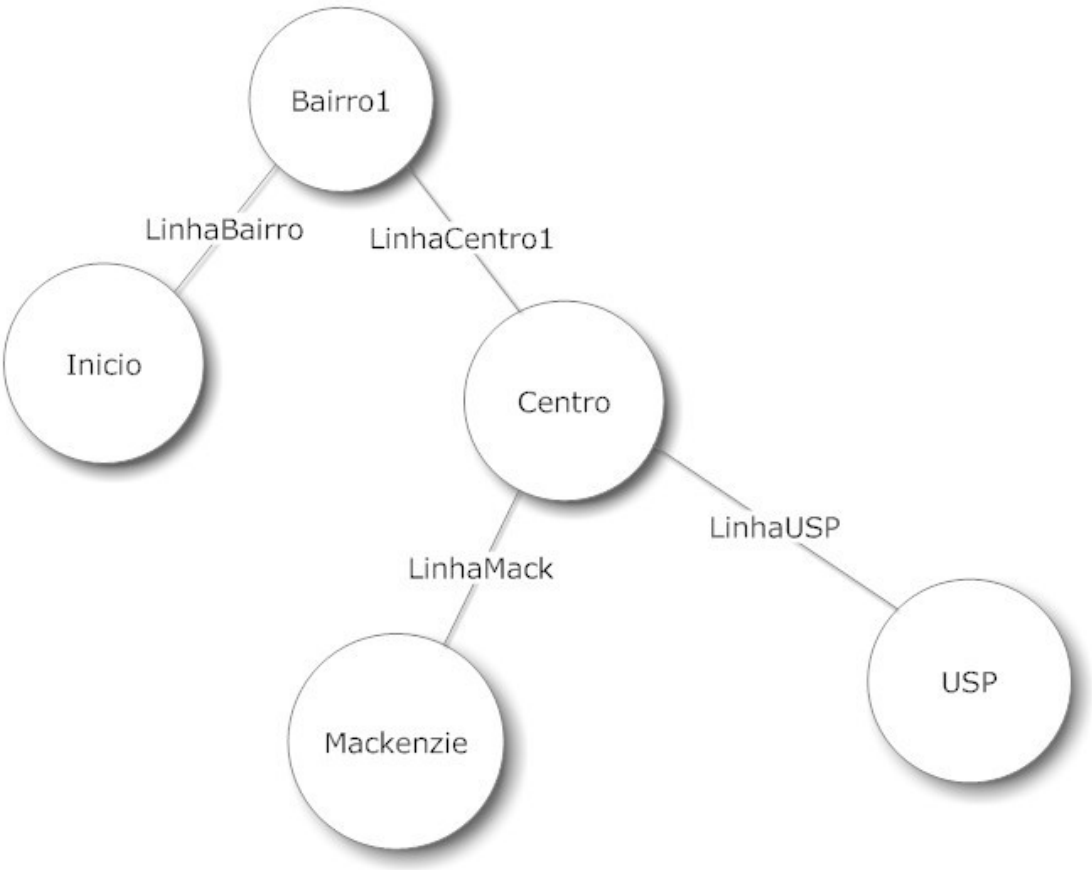

Figura 1 – Exemplo das Linhas de Metrô

Entrada

Os dados – fornecidos na entrada-padrão – contém apenas um caso de teste, que consiste nas informações do grafo e o caminho que se deseja encontrar, conforme o exemplo:

```
5 4
Inicio
Bairro1
Centro
Mackenzie
USP
0 1 LinhaBairro
1 2 LinhaCentro1
2 3 LinhaMack
2 4 LinhaUSP
0 4
```

A primeira linha de entrada contém dois números inteiros, **V** e **A**, que indicam respectivamente o número de Vértices e o número de Arestas do grafo.

As próximas **V** linhas indicam os rótulos dos vértices do grafo (**SV**).

As **A** linhas seguintes contém dois números inteiros **V1** e **V2** e uma string **SA** (sem espaços), que representam os vértices (**V1** e **V2**) que devem ser conectados por uma aresta no grafo e o rótulo da que os conecta aresta (**SA**).

A última linha da entrada indica o nó de início e o nó de fim do percurso.

Saída

A saída - apresentada do dispositivo padrão – consiste em apenas uma linha, com o seguinte formato:

<Vértice Origem>-[**<Rótulo aresta>**]-**<Vértice Intermediário 1>**-[**<Rótulo aresta>**]-**<Vértice Intermediário 2>...**-[**<Rótulo aresta>**]-**<Vértice Destino>**

Atenção: Não há espaços em branco na saída e nem quebras de linha.

Para a entrada apresentada no exemplo, a saída esperada é:

```
Inicio-[LinhaBairro]-Bairro1-[LinhaCentro1]-Centro-[LinhaUSP]-USP
```

Outras Informações Importantes

- O trabalho deve ser feito em duplas.
- O programa pode ser feito na linguagem C ou C++.
- Todas as submissões são checadas para evitar cópia/plágio/etc. Então, evite problemas e implemente o seu próprio código.
- Comente o seu código com uma explicação rápida do que cada função, método ou trecho importante de código faz (ou deveria fazer). Os comentários serão checados e valem nota.
- Mantenha a modularização entre código e dados, ou seja, divida o código explicitamente entre estruturas de dados (grafo) e algoritmos de manipulação. A modularização será checada e vale nota.
- Entradas/saídas devem ser lidas/escritas a partir dos dispositivos padrão, ou seja, use as funções "*printf(...)*" e "*scanf(...)*" / "*cin>>*" "*cout<<*". Para testar, arquivos podem ser redirecionados para/de seu programa na linha de comando utilizando os operadores < e >.
- Exemplo:

```
# ./trab0 < entrada.txt > saida.txt
```

Atenção

Fica a critério do aluno qual estrutura de dados utilizar para representar o dígrafo. Entretanto, para que as saídas dos programas sejam equivalentes entre a representação de matriz de adjacências e lista de adjacências, a visita dos vértices adjacentes ao vértice "i" deve ocorrer em **ordem crescente de índice:**
No caso da lista de adjacências, insira os vértices já ordenados;
No caso da matriz de adjacências, percorra a matriz em ordem crescente de índice.