

**ICMC – USP**  
**SCC603 – Algoritmos e Estruturas de Dados II**  
**Prof.<sup>a</sup> Rosane Minghim – 1º sem. 2012**

**Trabalho 2 - Betweenness**

Considere uma rede social / *microblog* similar ao Twitter (<http://www.twitter.com>), onde pessoas seguem e/ou são seguidas por outras pessoas. Nessa rede, além da funcionalidade básica de seguir outra pessoa, também é possível avaliar a participação dessa pessoa na rede, atribuindo uma nota de 1 a 5 estrelas. Tal rede pode ser modelada como um dígrafo – um grafo direcionado – onde os vértices são os participantes da rede e as arestas são suas relações direcionadas, ou seja:  $(u,v) = u$  segue  $v$ . As notas podem ser armazenadas como pesos nas arestas: se  $e = (u,v)$  então  $w(e) =$  nota de  $u$  para  $v$ .

O Trabalho 2 consiste no cálculo de uma medida de centralidade chamada *betweenness*. De uma forma geral, medidas de centralidade são usadas para calcular a importância dos vértices dentro do grafo. No caso de uma rede social, por exemplo, elas são usadas para calcular o grau de influência dos seus membros. Dentre as diversas medidas de centralidade existentes, *betweenness* é uma das mais utilizadas; o valor de *betweenness* de um vértice é relacionado à quantidade de menores caminhos do grafo que passam por ele.

O cálculo do valor de *betweenness* de um determinado vértice é feito da seguinte forma:

- a) Calcula-se todos os  $C$  menores caminhos entre todos possíveis pares de vértices do grafo.
- b) Dentre todos os  $C$  menores caminhos do grafo, o número  $C(v)$  indica a quantidade de menores caminhos que passam pelo vértice  $v$ . O vértice  $v$  não pode ser a *origem* ou o *destino* do caminho considerado.
- c) Consideraremos (neste trabalho) o valor de *betweenness* do vértice  $v$  como:

$$B(v) = C(v) / C$$

Ou seja, a relação entre o total de menores caminhos do grafo e a quantidade desses que passam por  $v$ . Esse número estará sempre entre 0 (nenhum menor caminho passa por  $v$ ) ou 1 (todos menores caminhos passam por  $v$ ).

Note que:

1. O menor caminho entre um vértice  $v$  e ele mesmo é  $[v]$ ; esse caminho faz parte de  $C$  mas não faz parte de  $C(v)$  (pois  $v$  é *origem* e *destino*).
2. Se um dos menores caminhos entre  $u$  e  $v$  é a aresta  $(u,v)$ , o caminho  $[u,v]$  faz parte de  $C$  mas não faz parte de  $C(u)$  ( $u$  é *origem*) nem  $C(v)$  ( $v$  é *destino*).

O arquivo de entrada consiste em:

```
20
nomeA nomeB:1 nomeC:4 nomeD:3
nomeFE nomeDF:3 nomeMS:2 nomeAXL:3
...
```

O grafo completo é dado como uma lista de adjacências. A primeira linha indica o número  $N$  de vértices do grafo. As  $N$  linhas subsequentes representam as relações de um participante da rede – *nome1* e *nome9*, no exemplo – separadas por **ESPAÇOS**; cada relação consiste no nome do seguido e na nota da relação, separados por dois pontos. Não há uma sequência pré-definida para as linhas dos participantes; a sequência é gerada aleatoriamente. Os nomes consistirão em **APENAS LETRAS, SEM NÚMEROS**, sem espaços ou quaisquer outros caracteres especiais. As notas serão inteiros de 1 a 5.

A saída deve ser gerada conforme o exemplo:

```
nomeA 0.118745  
nomeD 0.109454  
...
```

Cada linha deve conter o nome de um membro da rede com seu valor de *betweenness*, separados por espaço. Os membros devem ser classificados em ordem decrescente pelo valor de *betweenness*. O valor de *betweenness* deve ser impresso com 6 casas decimais. Um exemplo de comando de impressão que pode ser usado:

```
printf("%s %.6f\n", nome, betweenness);
```

**NOTE QUE PARA CADA PAR DE VÉRTICES PODE HAVER DIVERSOS MENORES CAMINHOS, OU SEJA, CAMINHOS COM VÉRTICES DIFERENTES MAS COM O MESMO VALOR MÍNIMO. TODOS DEVEM SER CONSIDERADOS NO CÁLCULO DO BETWEENNESS. OU SEJA, PARTINDO DE QUALQUER VÉRTICE S, NÃO TEREMOS NECESSARIAMENTE APENAS N-1 MENORES CAMINHOS PARA OS N-1 OUTROS VÉRTICES DO GRAFO; PODEM HAVER MAIS.**