

ICMC – USP
SCC603 – Algoritmos e Estruturas de Dados II
Prof.^a Rosane Minghim – 1º sem. 2012

Trabalho 1

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 1 consiste no cálculo dos menores caminhos entre pares de membros dessa rede. O arquivo de entrada fornecerá o grafo completo e um conjunto de pares de membros da rede. A saída deverá conter **TODOS** os menores caminhos entre cada par de membros fornecido na entrada.

Mais detalhadamente, o arquivo de entrada é dividido em duas partes:

```
20
nomeA nomeB:5 nomeC:2 nomeD:1
nomeD nomeGKI:2 nomeADF:2 nomeC:1
...
10
nomeA nomeC
nomeFR nomeDCV
...
```

Inicialmente é dado o grafo completo, 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 **ESPAÇO**; 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**, espaços ou quaisquer outros caracteres especiais. As notas serão inteiros de 1 a 5.

Em seguida são dados os pares de membros cujos menores caminhos devem ser calculados. A primeira linha indica o número M de pares que serão fornecidos. As M próximas linhas fornecem os pares de vértices.

A saída deve ser gerada conforme o exemplo:

```
nomeA nomeC 2
nomeA nomeD nomeC 2
nomeFR nomeD nomeBH nomeASZ nomeDCV 67
...
```

Cada linha deve conter uma descrição completa de **UM** menor caminho entre um par de vértices fornecido no arquivo de entrada, terminando com o valor total do custo do caminho descrito. Os pares devem aparecer na mesma sequência do arquivo de entrada. O custo é um valor inteiro (soma de todas notas do caminho).

NOTE QUE PARA CADA PAR DE VÉRTICES PODE HAVE DIVERSOS MENORES CAMINHOS, OU SEJA, CAMINHOS COM VÉRTICES DIFERENTES MAS COM O MESMO VALOR MÍNIMO. TODOS DEVEM SER GERADOS E IMPRESSOS. CAMINHOS DIFERENTES PARA UM MESMO PAR DEVEM SER IMPRESSOS EM ORDEM ALFABÉTICA (CRESCENTE), SIMILAR AO RESULTADO DA COMPARAÇÃO ABAIXO:

```
#include <string.h>

// caminho1 é "maior" que caminho2
if (strcmp(caminho1, caminho2) > 0) {
    printf("%s\n", caminho2);
    printf("%s\n", caminho1);
}
```