

Trabalho Prático 1

15 de setembro de 2014

1 O PROBLEMA DE RECOMENDAÇÕES DE AMIGOS

Você é o administrador principal do **Orkato**, uma das maiores redes sociais no Brasil. Um dos principais problemas que você tem no momento é a recomendação de amigos para outros usuários dentro da rede, e você teve uma ideia interessante para esse problema.

Você tem acesso ao banco de dados de todos os usuários e pode saber as relações de amizade entre eles. Para cada par de amigos você dispõe de duas métricas, a distância física entre os amigos e um valor que determina o nível de amizade deles (os dois são valores numéricos inteiros).

O objetivo deste trabalho é determinar se todos os usuários se encontram conectados, seja através de uma conexão direta ou indireta (i.e., através de outros amigos) e calcular qual seria o grafo que, conectando todos os usuários, maximiza a função de qualidade.

A função de qualidade é determinada pela soma total do nível de amizade pela soma das distâncias entre os amigos, como descrito a seguir na Equação 1.1:

$$Qualidade = \frac{\sum_{i=1}^E Friendship_i}{\sum_{i=1}^E Distance_i} \quad (1.1)$$

Definimos assim que a resolução pedida neste trabalho é achar um subgrafo da rede de amigos na rede social de modo a gerar o maior valor possível da

função de qualidade, seja por ligações diretas ou indiretas entre os usuários. É importante observar que nem todos os casos é possível realizar a ligação entre todos os usuários, deste o modo o algoritmo deve sinalizar essa impossibilidade retornando “-1.000”, como mostra o exemplo a seguir.

2 INSTRUÇÕES SOBRE IMPLEMENTAÇÃO

Serão aceitos somente implementações feitas em C, C++ e Java. Certifique-se que seu código pode ser executado e compilado em máquinas GNU/Linux. Os programas gerados devem receber as entradas diretamente da entrada principal (STDIN) e dar sua saída através da saída padrão (STDOUT), e somente o resultado da solução deve ser impresso, NADA ALÉM DISSO.

Além do código que resolve o problema proposto, você deverá escrever um arquivo em shell script (.sh) simples que compile o seu código e outro que o execute e submetê-los juntamente ao seu código com os nomes *compile.sh* e *execute.sh*. Segue o exemplo desses arquivos:

- **compile.sh**

```
#!/bin/bash
gcc main.c -o main
```

- **execute.sh**

```
#!/bin/bash
./main
```

No caso de executar o código feito em Java, pode ser usado o comando *java -jar tp1.jar*, onde *tp1.jar* é o executável do seu projeto gerado no *compile.sh*.

As implementações devem ser testadas em uma máquina do Departamento de Ciência da Computação de livre acesso aos alunos da pós-graduação via acesso remoto. Essa é a garantia de que a implementação será compilada e executada em um ambiente conhecido pelo aluno. Duas máquinas possíveis são, por exemplo:

- *cipo.grad.dcc.ufmg.br*
- *claro.grad.dcc.ufmg.br*

ENTRADA E SAÍDA

O programa devera solucionar múltiplas instâncias do problema em uma única execução. Cada instância do problema contém uma lista de usuários com o seguinte formato:

```
3
1 2 3 5
2 3 3 1
3 4 1 2
4 1 3 4
...
```

A primeira linha define a quantidade de usuários da rede. As linhas seguintes representam pares de amigos na rede e possuem 4 números inteiros separados por espaços, onde os dois primeiros números consistem no identificador de dois usuários que são amigos, o terceiro número define o nível da amizade e o quarto número define a distância entre os amigos.

Cada instância do problema é separada por uma linha vazia. A saída deve ser um número de ponto flutuante arredondado em 3 casas decimais para cada instância do problema seguindo a mesma ordem em que as instâncias aparecem na entrada.

Segue abaixo um exemplo que mostra as principais definições desse problema.

EXEMPLO

Entrada:	Saída:
4	
1 2 6 5	1.400
2 3 3 4	-1.000
3 4 5 2	0.706
4 1 3 3	
5	
1 2 6 5	
2 3 3 4	
3 4 5 2	
4 1 3 3	
4	
1 2 1 10	
2 3 3 3	
2 4 3 2	
3 4 3 1	
3 4 2 1	

3 DOCUMENTAÇÃO

Comece sua documentação com uma descrição breve do problema. Em seguida, apresente as sua solução para o problema proposto. Entre outros detalhes, é importante mencionar:

- Qual é o grafo usado, como ele é construído, qual problema é resolvido nesse grafo.
- As análises de complexidade de tempo e espaço de cada uma das soluções.
- Analise dos resultados obtidos.

- A documentação não deve exceder as 8 paginas.
- A documentação vai ser submetida via *minha.ufmg*.
- Data de entrega: **05/10**.

4 PARÂMETROS DE AVALIAÇÃO

A repartição dos pontos do trabalho pratico são:

- Modelagem do Problema (20%)
- Analise Teórica do Custo Assimptótico (10%)
- Execução e Analise de Experimentos (35%)
- Funcionamento Correto do Código (35%)

Boa sorte!