

# Trabalho II

## Matemática Discreta

O trabalho consiste na implementação de algoritmos de Teoria de Grafos.

1. Data da entrega: 19/03/2013. Trabalhos entregues após a data terão a nota consideravelmente reduzida.
2. Grupos: Os grupos podem ter até 4 alunos e os integrantes dos mesmos devem ser comunicados a mim até o dia 28/02/2013. Quem não possuir grupo até esta data realizará o trabalho sozinho.
3. Documentos a entregar:
  - Arquivo único no formato **.zip** contendo:
    - i. Os diretórios referentes a cada problema resolvido (cipó, engarraf, pedagio, redotica), contendo os demais arquivos.
    - ii. Os programas escritos nas linguagens C, C++ ou Java.
    - iii. Os arquivos de teste utilizados nos problemas.
    - iv. Um documento texto explicando como é realizada a execução de cada programa (compilação, execução e qualquer outro procedimento necessário).
    - v. Screenshots das submissões dos problemas para o Juiz Online <http://br.spoj.com/>. Como por exemplo:

ID		DATA	PROBLEMA	RESULTADO	TEMPO	MEM	LING
3948253	<input type="checkbox"/>	2010-08-21 23:59:48	Pedágio	aceito edit run	0.32	2.6M	C++ 4.3.2
3944394	<input type="checkbox"/>	2010-08-21 00:04:12	Pedágio	aceito edit run	0.30	2.6M	C++ 4.3.2
3944389	<input type="checkbox"/>	2010-08-21 00:03:26	Pedágio	aceito edit run	0.31	2.6M	C++ 4.3.2
3940040	<input type="checkbox"/>	2010-08-19 21:04:52	Pedágio	resposta errada edit run	0.00	2.6M	C++ 4.3.2
3939923	<input type="checkbox"/>	2010-08-19 20:39:05	Pedágio	resposta errada edit run	0.00	2.6M	C++ 4.3.2
3939702	<input type="checkbox"/>	2010-08-19 19:58:26	Pedágio	tempo limite excedido edit run	-	26M	C++ 4.3.2

4. Avaliação:
  - Entrega e organização de todos documentos.
  - Completude (o programa faz o que foi pedido).
  - Implementação e Documentação do programa.
  - Saída do problema impressa corretamente.
  - Submissão dos problemas ao Juiz Online: <http://br.spoj.com/>

#### 5. Validação:

- Todos os documentos entregues (exceto os arquivos de teste) devem conter o nome dos integrantes do grupo (coloque o nome dos integrantes no cabeçalho no arquivo).
- O arquivo **.zip** deve ser nomeado da seguinte maneira: **fulano-beltrano-cicrano-deltrano.zip**, onde 'fulano', 'ciclano', 'beltrano' e 'deltrano' correspondem aos nomes dos integrantes.
- Nenhum dos documentos apresentados poderá conter cópia, mesmo que parcial, de trabalho(s) de outro(s) grupo(s).
- Os problemas não devem possuir dependência de sistema operacional, ou seja, devem rodar em Windows, Linux ou Mac Os. Visto que os problemas possuem apenas operações de I/O triviais, esta restrição é facilmente obedecida.

#### 6. Descrição dos problemas:

- Para resolver os problemas utilizem os algoritmos de Dijkstra, Prim e Kruskal.
- As descrições dos problemas podem ser encontradas em anexo neste arquivo ou no site <http://br.spoj.com/> nos seguintes links:
  - i. <http://br.spoj.com/problems/PEDAGIO/>
  - ii. <http://br.spoj.com/problems/REDOTICA/>
  - iii. <http://br.spoj.com/problems/ENGARRAF/>
  - iv. <http://br.spoj.com/problems/CIPO/>
- **OBS:** O problema CIPO possui um erro na sua descrição: onde lê-se “**Você pode supor que existam cipós suficientes na selva para que sempre exista um sistema cipoviário que atenda os requisitos.**” substitua por: “**Você não pode supor que existam cipós suficientes na selva para que sempre exista um sistema cipoviário que atenda os requisitos.**”

#### 7. Observações:

- Você deve cadastrar-se no site <http://br.spoj.com/> para que possa enviar os problemas para submissão.
- Após realizado o cadastro, para submeter um problema você deve acessar a página do problema (links disponíveis no item 6), clicar em **Submeter**, selecionar o arquivo com o **código fonte**, e **selecionar a Linguagem** na qual o programa foi escrito, e então clicar em **Enviar**. Aguarde a resposta do Juiz Online.
- Você pode utilizar os casos de teste fornecidos pelos problemas para testar seu programa. Para isso, salve os dados de entrada em um arquivo de texto (.txt) e rode o programa da seguinte maneira: **programa.exe < entrada.txt**

Desta maneira o programa irá ler o arquivo como se fosse a entrada padrão (entrada pelo teclado).

Para criar um arquivo com a saída do seu programa você pode fazer da seguinte maneira: **programa.exe > saida.txt**

Assim, o programa irá criar um arquivo texto com o conteúdo da saída padrão (monitor).

Você pode usar um arquivo de entrada e gerar um arquivo de saída simultaneamente: **programa.exe < entrada.txt > saida.txt**

Estes comandos são análogos para Linux/Mac OS:

**./programa < entrada.txt**

**./programa > saida.txt**

**./programa < entrada.txt > saida.txt**

## SPOJ Problem Set (seletivas)

### 2840. Engarrafamento

#### Problem code: ENGARRAF

*Autor: Tiago Mota*

Cansado dos constantes engarrafamentos que enfrenta todo dia para chegar ao trabalho com seu carro, Antônio resolveu reclamar com a secretaria responsável pelo controle de tráfego de sua cidade. Porém, ao entrar em contato com tal órgão, descobriu que este disponibiliza mapas das ruas da cidade, com o tempo que um carro costuma gastar para atravessar cada uma das ruas durante o período de engarrafamento. Com estas informações, ele pôde descobrir o caminho mais rápido de sua casa até seu local de trabalho, podendo, assim, acordar mais tarde e dormir um pouco mais todo dia.

Empreendedor nato, Antônio percebeu que as informações colhidas junto a secretaria poderiam ser úteis para ele de outra forma. Decidiu, então, criar um programa de computador para ajudar seus colegas de trabalho que enfrentam o mesmo problema, indicando os caminhos mais rápidos entre pontos da cidade selecionados pelo usuário. Naturalmente, esta ajuda não será gratuita, já que Antônio pretende cobrar pelo uso do programa.

Antônio, porém, não tem tempo livre suficiente para programar. Como você é um estagiário da empresa que está logo abaixo de Antônio, ele pede que você escreva um programa que, dado um mapa com as ruas e o tempo necessário para atravessar cada uma, calcule o tempo mínimo para se ir de um determinado local de origem a um determinado local de destino.

#### Entrada

A entrada é composta de vários casos de teste. Cada caso de teste é iniciado com uma linha contendo dois números inteiros,  $n$  e  $m$  ( $1 \leq n \leq 100$ ,  $0 \leq m \leq 10000$ ), representando, respectivamente, o número de locais da cidade (identificados por inteiros de 1 a  $n$ ) e o número de ruas a serem considerados.

A seguir, há  $m$  linhas, cada uma contendo três inteiros,  $i$ ,  $j$  e  $t$  ( $1 \leq i, j \leq n$ ,  $i \neq j$ ,  $1 \leq t \leq 1000$ ), indicando que há uma rua de mão única ligando os locais  $i$  e  $j$ , nesta ordem, e cujo tempo de travessia é  $t$ . Para cada par de locais  $i$  e  $j$ , haverá no máximo uma rua ligando  $i$  a  $j$  e no máximo uma rua ligando  $j$  a  $i$ .

Finalmente, a última linha do caso de teste contém dois inteiros,  $s$  e  $d$  ( $1 \leq s, d \leq n$ ), respectivamente, os locais inicial e final para os quais devemos calcular o tempo mínimo de trajeto.

A entrada termina quando  $n = m = 0$ .

#### Saída

Para cada caso de teste, seu programa deve imprimir uma linha com um único inteiro, representando o tempo mínimo necessário para se deslocar do local  $s$  ao local  $d$ . Se não for possível alcançar o local  $d$  a partir do local  $s$ , imprima  $-1$  como resposta.

## Exemplo

### Entrada:

```
3 3
1 2 2
1 3 7
2 3 3
1 3
3 3
1 2 3
1 3 7
2 3 5
1 3
3 2
1 2 2
2 1 3
1 3
4 6
1 2 5
1 3 3
2 4 3
3 2 2
4 1 4
4 3 9
2 3
0 0
```

### Saída:

```
5
7
-1
10
```

---

Added by: Wanderley Guimaraes

Date: 2008-07-08

Time limit: 1s

Source limit:50000B

Languages: All

Resource: Segunda prova de TEP - 2008/1 - UFRJ

# SPOJ Problem Set (obi)

## 819. Pedágio

### Problem code: PEDAGIO

Como premio pela primeira colocação na Olimpíada Brasileira de Informática, Juquinha e sua família ganharam uma viagem de uma semana a Coréia do Sul. Como o país é deslumbrante, com tradições, cultura, arquitetura e culinária muito diferentes das do Brasil, o pai de Juquinha, o Sr. Juca, decidiu alugar um carro para conhecer melhor o país. As estradas são muito bem cuidadas; todas são de sentido duplo, e duas cidades podem ser ligadas diretamente por mais de uma estrada. No entanto, em todas as estradas paga-se um pedágio de valor fixo (há um pedágio em cada direção, entre duas cidades). Como o Sr. Juca não tem muito dinheiro para gastar, as viagens com o carro devem ser muito bem planejadas.

### Tarefa

Escreva um programa que, conhecidas as cidades e estradas existentes no país, e a cidade onde Juquinha e sua família estão, encontre cada cidade (que não a cidade onde eles estão) que possa ser visitada por eles, dada a restrição de que o Sr. Juca deseja pagar no máximo  $P$  pedágios (considerando apenas a viagem de ida).

### Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém quatro números inteiros  $C$ ,  $E$ ,  $L$  e  $P$ . Os valores  $C$  e  $E$  indicam respectivamente o número de cidades e o número de estradas existentes. As cidades são identificadas por inteiros de 1 a  $C$ . Os valores  $L$  e  $P$  indicam, respectivamente, a cidade onde a família de Juquinha está no momento e o número máximo de pedágios que o Sr. Juca está disposto a pagar. As  $E$  linhas seguintes contêm cada uma a informação de uma estrada, representada por um par de números inteiros positivos  $X$  e  $Y$ , indicando que há uma estrada (de sentido duplo) da cidade  $X$  para a cidade  $Y$ . O final da entrada é indicado por  $C = E = L = P = 0$ .

### Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste  $n$ ", onde  $n$  é numerado a partir de 1. Na segunda linha devem aparecer os identificadores das cidades que podem ser alcançadas, em ordem crescente, separados por pelo menos um espaço em branco. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

### Exemplo

**Entrada:**

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

```
9 12 1 2
2 1
1 5
2 1
3 2
9 3
3 4
4 8
4 7
7 6
5 6
4 5
3 7
0 0 0 0
```

**Output:**

```
Teste 1
1 3
```

```
Teste 2
2 3 4 5 6
```

## Restrições

$0 \leq C \leq 50$  ( $C = 0$  apenas para indicar o fim da entrada)  
 $0 \leq E \leq 2500$  ( $E = 0$  apenas para indicar o fim da entrada)  
 $0 \leq L \leq C$  ( $L = 0$  apenas para indicar o fim da entrada)  
 $0 \leq P \leq C$  ( $P = 0$  apenas para indicar o fim da entrada)  
 $1 \leq X \leq C$   
 $1 \leq Y \leq C$

---

Added by: Wanderley Guimaraes

Date: 2006-04-21

Time limit: 1s

Source limit: 50000B

Languages: All

Resource: Olimpiada Brasileira de Informatica 2002

## SPOJ Problem Set (obi)

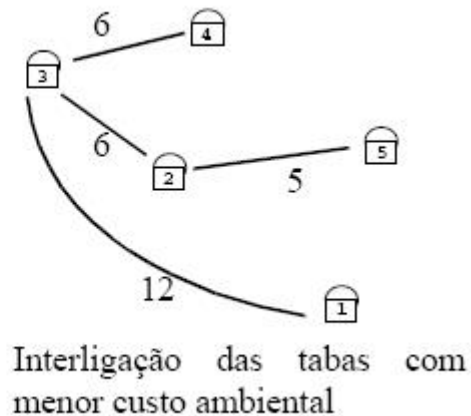
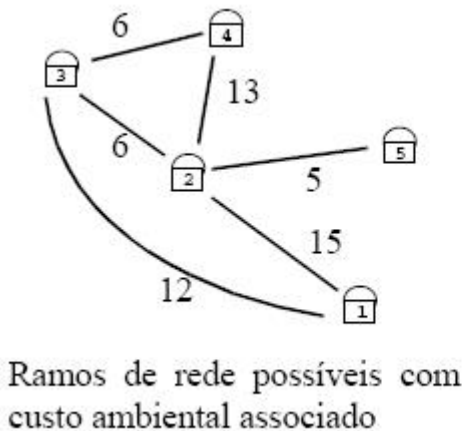
### 810. Rede ótica

#### Problema: REDOTICA

Os caciques da região de Tutuaçu pretendem integrar suas tribos à chamada “aldeia global”. A primeira providência foi a distribuição de telefones celulares a todos os pajés. Agora, planejam montar uma rede de fibra ótica interligando todas as tabas. Esta empreitada requer que sejam abertas novas picadas na mata, passando por reservas de flora e fauna. Conscientes da necessidade de preservar o máximo possível o meio ambiente, os caciques encomendaram um estudo do impacto ambiental do projeto. Será que você consegue ajudá-los a projetar a rede de fibra ótica?

#### Tarefa

Vamos denominar uma ligação de fibra ótica entre duas tabas de um ramo de rede. Para possibilitar a comunicação entre todas as tabas é necessário que todas elas estejam interligadas, direta (utilizando um ramo de rede) ou indiretamente (utilizando mais de um ramo). Os caciques conseguiram a informação do impacto ambiental que causará a construção dos ramos. Alguns ramos, no entanto, nem foram considerados no estudo ambiental, pois sua construção é impossível.



Sua tarefa é escrever um programa para determinar quais ramos devem ser construídos, de forma a possibilitar a comunicação entre todas as tabas, causando o menor impacto ambiental possível.

#### Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros positivos  $N$  e  $M$  que indicam, respectivamente, o número de tabas e o número de ramos de redes possíveis. As tabas são numeradas de 1 a  $N$ . As  $M$  linhas seguintes contêm três inteiros positivos  $X$ ,  $Y$  e  $Z$ , que indicam que o ramo de rede que liga a taba  $X$  à taba  $Y$  tem impacto ambiental  $Z$ . Com os conjuntos de teste dados sempre é possível interligar todas as tabas. O final da entrada é indicado quando  $N = 0$ .



## Saída

Para cada conjunto de teste da entrada seu programa deve produzir uma lista dos ramos de redes que devem ser construídos. A lista deve ser precedida de uma linha que identifica o conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A lista é composta por uma sequência de ramos a serem construídos, um ramo por linha. Um ramo é descrito por um par de tabas X e Y, com  $X < Y$ . Os ramos de rede podem ser listados em qualquer ordem, mas não deve haver repetição. Se houver mais de uma solução possível, imprima apenas uma delas. O final de uma lista de ramos deve ser marcado com uma linha em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

## Exemplo

### Entrada:

```
3 3
1 2 10
2 3 10
3 1 10
5 6
1 2 15
1 3 12
2 4 13
2 5 5
3 2 6
3 4 6
0 0
```

### Saída:

```
Teste 1
1 2
1 3
```

```
Teste 2
```

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

## Restrições

$0 \leq N \leq 100$  ( $N = 0$  apenas para indicar o fim da entrada)

$1 \leq M \leq N(N-1)/2$

$1 \leq X \leq 100$

$1 \leq Y \leq 100$

$1 \leq Z \leq 100$

---

Adicionado por: [Wanderley Guimarães](#)

Data: 2006-04-19

Tempo limite: 1s

Tamanho do fonte: 50000B

Linguagem permitida: Todas exceto: CLOJ ERL F# GO JS SCALA TCL TECS

Origem: Olimpíada Brasileira de Informática 2000

## SPOJ Problem Set (seletivas)

### 1746. Sistema Cipoviário

#### Problem code: CIP0

Os pesquisadores do departamento de pesquisa operacional da Universidade da Columbia Britânica foram contratados para uma estranha tarefa. Vários países da África resolveram se unir e utilizar oficialmente o meio de transporte que ficou mundialmente conhecido nos filmes do Tarzan: o cipó. Há milhões de cipós na África e é surpreendente com que velocidade e eficiência uma pessoa pode se deslocar na selva utilizando esse meio de transporte. Só surgiu um pequeno problema. Os cipós são dominados por três grandes tribos: os makeleles, os malouhdás e os abedis. As tribos exigem ser pagas por cipó usado no sistema de transporte. Como eles ainda não sabem o significado de palavras como cartel, cada uma fez o seu preço, e divergiram bastante. Enquanto os makeleles exigem 1235 bongôs por cipó usado, os malouhdás exigem 8977 e os abedis 10923 (a Jane ainda está viva, e ajudou a intermediar a negociação para esta tribo).

Os pesquisadores foram contratados para escolher os cipós que comporão o primeiro sistema cipoviário do mundo. Os contratantes construíram milhões de "pontos de cipó" pela selva africana e desejam que os cipós sejam escolhidos de tal forma que seja possível ir de qualquer ponto a qualquer outro usando os cipós contratados (você pode ter de trocar de cipó algumas vezes, como fazia o Tarzan). Você deve dizer qual o custo de um sistema que atenda estes requisitos e seja o mais barato possível.

Você pode supor que existam cipós suficientes na selva para que sempre exista um sistema cipoviário que atenda os requisitos.

#### Entrada

A entrada é composta de diversas instâncias. A primeira linha de cada instância contém dois inteiros  $n$  ( $1 \leq n \leq 1000$ ) e  $m$  ( $1 \leq m \leq 2000000$ ), onde  $n$  é o número de "pontos de cipó" e  $m$  é o número de cipós. Cada uma das  $m$  linhas seguintes contém três inteiros  $u$ ,  $v$  e  $c$  indicando que existe um cipó que vai do ponto  $u$  e até o ponto  $v$  com custo  $c$ , onde  $1 \leq u, v \leq n$  e  $c = 1235$  ou  $8977$  ou  $10923$ .

A entrada termina com final de arquivo.

#### Saída

Para cada instância, você deverá imprimir um identificador `Instancia k`, onde  $k$  é o número da instância atual. Na linha seguinte imprima o custo de um sistema que atenda os requisitos descritos acima.

Após cada instância imprima uma linha em branco.

## Exemplo

### Entrada:

```
3 3
1 2 10923
1 3 1235
2 3 1235
3 2
1 2 1235
2 3 10923
```

### Saída:

```
Instancia 1
2470
```

```
Instancia 2
12158
```

---

Added by: Wanderley Guimaraes

Date: 2007-08-28

Time limit: 9s

Source limit:50000B

Languages: All

Resource: Seletiva para Maratona de Programação do IME - 2007