



## SCC0203 - Algoritmos e Estruturas de Dados II

Trabalho Prático – Unidade 1 – Grafos

**Prazo máximo: 23:55h de 22/04/2012**

A seguir são apresentados três problemas: 1) Lista de disciplinas; 2) Rede de distribuição de água; e 3) Despachantes certificados. Como trabalho prático, você deve escolher **dois** desses três **problemas** para implementar uma solução. Em todos os casos os problemas devem ser modelados utilizando grafos e deve ser buscada a solução mais eficiente. Apenas a linguagem de programação **C (padrão ANSI)** deverá ser utilizada. O trabalho deverá ser desenvolvido **individualmente**.

### Testes

Inicialmente as implementações podem ser testadas utilizando o caso de teste apresentado como exemplo em cada projeto. A partir do dia 18/04/2012, os três projetos estarão disponível no sistema de submissão de trabalhos do Moodle, com os casos de teste oficiais. Em cada projeto, deve ser submetido um **único arquivo .c** contendo o código fonte da solução do problema, incluindo os TADs. Além disso, os padrões de entrada e saída devem ser seguidos **rigorosamente**. Espaços desnecessários ou linhas em branco **podem** ser acusados como erro de apresentação. É suficiente a utilização das funções `scanf()` e `printf()`, respectivamente, para leitura e escrita nos dispositivos de entrada e saída padrões.

### Prazos

Você terá **no máximo** até às 23:55h do dia 22/04/2012 para realizar submissões no sistema. Submissões fora do prazo não serão aceitas. No período em que o sistema estiver aberto poderão ser realizadas tantas submissões quanto forem necessárias. Lembre-se, no entanto, que é necessário excluir o arquivo da submissão anterior para que possa ser realizada uma nova submissão.

### Avaliação

Para fins de avaliação, apenas a última submissão de cada problema será considerada. As soluções dos dois problemas onde forem obtidas as maiores pontuações serão avaliadas. Será analisado o código fonte, especialmente os TADs utilizados. Os trabalhos também serão inspecionados quanto a plágio, tanto entre trabalhos submetidos, como em relação à internet.



## SCC0203 - Algoritmos e Estruturas de Dados II

### Projeto 1 – Lista de disciplinas

Um dos problemas na coordenação de um curso de graduação é a formulação de sua grade curricular e a distribuição de suas disciplinas em semestres. É necessário, para isso, considerar dependências entre disciplinas e também equilibrar a carga horária em cada semestre. Adicionalmente, devem-se evitar sequências muito longas de dependências entre disciplinas, pois isto pode ocasionar o atraso demorado do curso de alunos reprovados. Como coordenador do curso de ciência da computação, o Prof. Raimundo resolveu que uma forma interessante de gerenciar esse problema seria, além de considerar as dependências entre as disciplinas, considerar também o índice de reprovação médio das turmas em cada disciplina. Desse modo, o Prof. Raimundo acredita que será possível ordenar as disciplinas tal que as disciplinas mais difíceis (com índices maiores de reprovação) sejam cursadas o quanto antes e as sequências de dependência sejam reduzidas.

Como aluno de Alg. 2, o Prof. Raimundo escolheu você para implementar a sua ideia. Escreva um programa que, dado um conjunto de disciplinas e os respectivos índices de reprovação, e um conjunto de dependências entre disciplinas, identifique, segundo a solução do Prof. Raimundo, a ordem (linear) mais adequada de as disciplinas serem cursadas. Não é necessário, no entanto, dividi-las em semestres. Considere que jamais existirão mais do que 100 disciplina em uma grade curricular.

### **Entrada**

A primeira e segunda entrada é dada, respectivamente, pelo número de disciplinas,  $N$ , e pelo número de dependências entre disciplinas,  $M$ . Cada uma das  $N$  linhas de entradas subsequentes é composta pelo nome da disciplina e pelo respectivo índice de reprovação. Em seguida, cada uma das  $M$  linhas de entradas subsequentes é composta pelo nome de uma disciplina que possui dependência e o respectivo nome de sua dependência (pré-requisito). Nomes de disciplinas estão limitados a 20 caracteres. Nomes compostos, por sua vez, são conectados através do caractere “\_”.

### **Saída**

Como saída deve ser exibida a lista de disciplinas ordenadas segundo o critério proposto pelo Prof. Raimundo. Cada linha deve exibir o nome de uma disciplina.

## Exemplo

Como foi observado que o problema poderia ser interpretado de duas (e somente duas) maneiras, a seguir é apresentado um exemplo de entrada e duas saídas (Solução 1 e Solução 2) do programa. A partir do dia 18/04 estará disponível no sistema de submissão de trabalhos do Moodle os testes “Projeto 1 – Lista de disciplinas (Solução 1)” e “Projeto 1 – Lista de disciplinas (Solução 2). Lembre-se que o padrão de saída deve ser seguido rigorosamente.

### Entrada

```
10 10
Calculo1 63
Calculo2 50
Calculo3 30
GA 42
Fisical 40
Fisica2 60
ICC1 12
ICC2 26
ALG1 14
ALG2 2
Calculo2 Calculo1
Calculo2 GA
Calculo3 Calculo2
Fisical Calculo1
Fisica2 Calculo2
Fisica2 Fisical
ICC2 ICC1
ALG1 ICC1
ALG2 ALG1
ALG2 ICC2
```

### Saída (Solução 1)

```
Calculo1
GA
Calculo2
Fisical
Fisica2
Calculo3
ICC1
ICC2
ALG1
ALG2
```

### Saída (Solução 2)

```
Calculo1
GA
ICC1
Calculo2
Fisical
ICC2
ALG1
Fisica2
Calculo3
ALG2
```



## SCC0203 - Algoritmos e Estruturas de Dados II

### Projeto 2 – Rede de distribuição de água

Todos os dias em Chico City as pessoas consomem em suas residências dezenas e dezenas de litros de água. Porém, grande parte desse consumo é desperdiçado. Embora não se preocupem com o consumo desenfreado de água, basta uma falha no sistema de distribuição para que os cidadãos de Chico City retomem os protestos às portas da prefeitura do município. Muito chateado com as críticas do povo, o prefeito Walfrido Canavieira decidiu ampliar a rede de distribuição de água do município. A fim de suportar situações de contingência, ele decidiu que deveriam ser criadas redundâncias no sistema de distribuição de água interligando fisicamente os pontos de armazenamento de cada subprefeitura. Cada um dos subprefeitos de Chico City foi solicitado para preparar um mapa com a localização de todos os pontos de armazenamento de sua subprefeitura.

No entanto, como era de se esperar, o orçamento municipal ainda está no vermelho. Por isso, o prefeito Canavieira decidiu firmar um contrato de estágio com você para que escreva um programa capaz de identificar a maneira mais eficiente de interligar os pontos de armazenamento em uma subprefeitura, minimizando os custos com infraestrutura. Nenhuma subprefeitura possui mais que 40 pontos de armazenamento. Cada conexão entre pontos de armazenamento, por sua vez, deve ser construída de forma que o fluxo de água ocorra em ambos os sentidos. O comprimento do encanamento é dado pela distância entre respectivos dois pontos de armazenamento.

### **Entrada**

Por vez, cada subprefeito irá fornecer a localização de todos os pontos de armazenamento do sistema de distribuição de água da sua subprefeitura. A primeira entrada informa o número de pontos de armazenamento,  $N$ . Cada uma das  $N$  linhas subsequentes apresenta duas entradas com as coordenadas de cada ponto de armazenamento de acordo com o mapa.

### **Saída**

Como saída devem ser apresentadas as conexões que devem ser realizadas entre os pontos de armazenamento. Use para isso uma matriz booleana  $N \times N$ .

## Exemplo

A seguir é apresentado um exemplo de entrada e saída do programa. A partir do dia 18/04 estará disponível no sistema de submissão de trabalhos do Moodle o teste “Projeto 2 – Rede de distribuição de água”. Lembre-se que o padrão de saída deve ser seguido rigorosamente.

### *Entrada*

```
4
9.2 10.1
2.4 5.8
3.1 8.5
5 6.4
```

### *Saída*

```
0 0 0 1
0 0 1 1
0 1 0 0
1 1 0 0
```



## SCC0203 - Algoritmos e Estruturas de Dados II

### Projeto 3 – Despachantes certificados

Cansada de tanto mau trato, D. Sofia finalmente resolveu dar o troco no Seu Nazareno. Ela embarcou numa viagem de 6 meses a fim de conhecer o Brasil. Seu Nazareno, por sua vez, teve que arcar com todas as suas despesas. Ficou responsável de, todas as semanas, depositar uma quantia em sua conta do Chico City Bank. O que os dois não esperavam é que apenas em Chico City existissem agências do Chico City Bank. Com isso, para que a expedição de D. Sofia não seja cancelada, seu Nazareno teve a ideia de procurar alguns despachantes certificados a fim de enviar dinheiro à D. Sofia. Contudo, os problemas não param por aí. Devido a uma restrição do sindicato de classe, um despachante pode atuar no máximo em 8 cidades. Além disso, cada despachante cobra um valor específico por trecho entre cidades, quantia que é descontada do montante a ser enviado ao destino. Por isso, a fim de que seja deduzido o mínimo possível do dinheiro enviado à D. Sofia, o Seu Nazareno teria que identificar dentre todos os despachantes e cidades em que atuam a cadeia de despachantes e o trajeto mais barato de Chico City até a cidade em que a D. Sofia estará visitando.

Como Seu Nazareno não é muito instruído, ele pede encarecidamente a você que o ajude. Ele precisa que você elabore um programa que, dadas algumas listas de trechos e preços de alguns despachantes, seja identificada a cadeia de cidades e respectivos despachantes que seja a mais econômica. Considere, adicionalmente, que o Seu Nazareno não se daria ao trabalho de pesquisar preços em mais que 3 despachantes.

### **Entrada**

A primeira e segunda entradas são dadas, respectivamente, pelo nome da cidade visitada por D. Sofia, e pelo número de despachantes certificados pesquisados por seu Nazareno,  $N$ . Para cada um dos  $N$  despachantes existe uma linha onde é dado como entrada, primeiramente o nome, e em seguida o número de trechos cobertos pelo respectivo despachante,  $M$ . Cada um dos  $M$  trechos de cada despachante e seu preço associado é dado por 3 entradas, em cada uma das  $M$  linhas subsequentes. Nomes de cidades e despachantes estão limitados a 30 caracteres. Nomes compostos são conectados através do caractere “\_”. Nomes de cidades são apresentados seguindo um mesmo padrão independentemente do despachante.

## Saída

Para cada caso de teste deve ser exibido como saída o valor total dos serviços (com precisão de duas casas decimais), a cidade de origem, e uma lista (ordenada) com os pares de despachantes e cidades que compõem o trajeto correspondente. Caso não seja possível identificar uma cadeia de cidades e despachantes, deve-se exibir o valor -1 como saída.

## Exemplo

A seguir é apresentado um exemplo de entrada e saída do programa. A partir do dia 18/04 estará disponível no sistema de submissão de trabalhos do Moodle o teste “Projeto 3 – Despachantes certificados”. Lembre-se que o padrão de saída deve ser seguido rigorosamente.

### Entrada

```
Ipeuna 3
Vai_e_Vem_SC 6
Ipeuna Analandia 6.40
Sao_Carlos Analandia 8.80
Sao_Carlos Campinas 12.30
Sao_Carlos Itirapina 6.20
Sao_Carlos Sao_Paulo 14.20
Analandia Itirapina 3.20
Chico_City_DC 2
Chico_City Sao_Paulo 32.20
Chico_City Campinas 33.10
Rio_Claro_DC 5
Ipeuna Analandia 6.20
Rio_Claro Analandia 10.20
Rio_Claro Campinas 10.60
Rio_Claro Itirapina 6.20
Rio_Claro Sao_Carlos 8.10
```

### Saída

```
59.30
Chico_City
Chico_City_DC
Campinas
Rio_Claro_DC
Rio_Claro
Rio_Claro_DC
Itirapina
Vai_e_Vem_SC
Analandia
Rio_Claro_DC
Ipeuna
```