UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
http://www.icmc.usp.br
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÂO

SCC0216 – Modelagem Computacional em Grafos
Prof.ª Rosane Minghim

ROTEIRO DE LABORATORIO

Informações Gerais

- A seguir são apresentados dois problemas que devem ser trabalhados **individualmente** ao longo desse período de laboratório: 1) Classificação de arestas; e 2) Sistema de distribuição de águas. Em ambos os casos o problema deve ser modelado utilizando grafos e deve ser buscada a solução mais eficiente. Mantenha o código organizado fazendo uso de boas práticas de programação.

- Ao chegar a uma solução satisfatória, submeta-a no Sistema de Submissão de Programas (SSP), disponível no endereço **`ssp.icmc.usp.br`**, para que possa ser validada em alguns casos de teste. Os padrões de entrada e saída devem ser seguidos **rigorosamente**.

- O SSP aceita submissões de um único arquivo fonte (.c ou .cpp) ou de um arquivo .zip com fontes e bibliotecas e um arquivo **`Makefile`** com as rotinas de compilação do programa. Mais detalhes estão disponíveis no documento em **`ssp.icmc.usp.br/SSP.pdf`**.

- Entradas e saídas devem ser lidas e escritas a partir dos dispositivos de entrada e saída padrões, logo, não é necessária a manipulação de arquivos, e são suficientes a utilização das funções `scanf()`e `printf()` da biblioteca stdio do C ou os objetos e operadores `cin >>` e `cout <<` da biblioteca iostream do C++.

- Para testar o programa fora do SSP, pode-se usar redirecionamento de arquivos. Para isso utilize os operadores **<** e **>**, como no seguinte exemplo:

```
$ ./exercicio < entrada.txt > saida.txt
```

- O SSP estará aberto para a submissão desses exercícios apenas durante o período deste laboratório, das 13:20 às 16:00 hs. Se você obtiver sucesso **em todos os casos de teste**, será somado 1 ponto adicional à nota do trabalho da unidade.

- Encerrados os prazos para a entrega do Trabalho 2, serão disponibilizados, na página da disciplina na Wiki e no SSP, a descrição do trabalho da unidade e os casos de teste.

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
http://www.icmc.usp.br
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÂO

SCC0216 – Modelagem Computacional em Grafos
Prof.ª Rosane Minghim

ROTEIRO DE LABORATORIO

Despachantes certificados

Cansada de tanto mau trato, D. Sofia finalmente resolveu dar o troco no Seu Nazareno. Ela embarcou numa viajem 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 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. 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
```

Sugestão de roteiro de atividades:

1. Crie um TAD grafo. Funções como inserção de aresta e definição do número de vértices do grafo são essenciais. Note que mais de que um despachante pode ser responsável por um trecho.

2. Implemente as funções e estruturas necessárias à leitura e armazenamento das entradas do programa. As funções **`scanf()`** e **`printf()`** da linguagem C são suficientes para a leitura e escrita nos dispositivos de entrada e saída padrões.

3. Implemente um algoritmos para a identificação do caminho de custo mínimo adequado à solução do problema. Pode-se usar o TAD Heap, disponível na página da disciplina na Wiki.

4. Modele a saída do programa de acordo com o padrão estabelecido.