

Tarefa Nº 03 - Viação Grafeira

Prazo de entrega: **Consultar a página da tarefa.**

Linguagem para implementação: **C++**.

Professor: Andrei Braga

Viação Grafeira

Considere o seguinte problema. A empresa de transporte rodoviário de passageiros Viação Grafeira está planejando uma reformulação da sua malha de trechos de viagens. Neste processo, a partir de algumas cidades de origem, a empresa quer determinar quais cidades de destino são economicamente viáveis de serem mantidas em atividade. Neste contexto, **economicamente viável** significa que a extensão total mínima de uma sequência de trechos da cidade de origem para a de destino é maior ou igual a um valor mínimo definido pela empresa – caso contrário, não haverá muitos passageiros porque as pessoas vão preferir viajar de carro particular.

Nesta tarefa, você deve implementar uma classe que represente um grafo **dirigido** (digrafo) **simples** que possua **pesos não-negativos nas arestas**. O digrafo deve ser representado como **listas de adjacência**. Você deve escrever um método que executa no digrafo uma operação que resolve o problema descrito acima com base no **Algoritmo de Dijkstra**.

Você deve escrever um programa que constrói um digrafo, executa operações no digrafo e depois, se necessário, explicitamente o destrói. O seu programa deve processar informações que determinarão as operações a serem executadas no digrafo, o que deve ser feito de acordo com as **Seções Entrada e Saída** abaixo.

Entrada

A primeira linha da entrada contém dois inteiros **C** ($C > 0$) e **T** ($T \geq 0$), sendo **C** o número de cidades atendidas na malha de viagens da empresa e **T** o número de trechos da malha. Cada uma das **T** linhas seguintes contém três inteiros **X**, **Y** e **Z**, indicando que a malha de viagens da empresa contém um trecho da cidade **X** para a cidade **Y** com extensão **Z**.

A próxima linha da entrada contém um inteiro **R** ($R \geq 0$), que determina para quantas cidades de origem a empresa vai reavaliar quais cidades de destino são economicamente viáveis de serem mantidas em atividade. Cada uma das **R** linhas seguintes contém um inteiro **X** e um inteiro **M**, indicando que a empresa vai fazer a reavaliação relativa à cidade **X** considerando o valor mínimo de extensão **M**.

Saída

A saída deve consistir no seguinte:

- Para cada reavaliação a ser feita pela empresa relativa à cidade **x**, o seu programa deve imprimir **C - 1** linhas, uma para cada cidade **u** diferente de **x**, em ordem crescente dos seus índices. Cada uma destas linhas deve conter
 - **u** seguido do caractere **:** e de um espaço em branco e,
 - caso a extensão total mínima de uma sequência de trechos de **x** para **u** seja maior ou igual ao valor mínimo definido,

- esta extensão seguida do caractere , e
- os índices das cidades de uma sequência de trechos de **x** para **u** de extensão total mínima – os índices das cidades devem ser impressos antecedidos por um espaço em branco e na ordem em que aparecem na sequência de trechos **no sentido** de **u** para **x**;
- caso a extensão total mínima de uma sequência de trechos de **x** para **u** seja menor que o valor mínimo definido ou caso não exista uma sequência como esta,
 - o texto a seguir:
economicamente inviavel partindo de **x**
- Além disso, para cada reavaliação a ser feita pela empresa, após as impressões acima, o seu programa deve imprimir uma linha em branco (inclusive após a última reavaliação).

Exemplos de execução

| Entrada | Saída |
|--|--|
| 4 5 0 1 6 0 2 2 2 3 3 3 0 4 3 1 2 1 0 3 | 1: 6, 1 0 2: economicamente inviavel partindo de 0 3: 5, 3 2 0 |

| Entrada | Saída |
|---|--|
| 4 5 1 0 2 1 3 3 2 1 1 2 3 5 3 1 4 2 2 3 0 3 | 0: 3, 0 1 2 1: economicamente inviavel partindo de 2 3: 4, 3 1 2 1: economicamente inviavel partindo de 0 2: economicamente inviavel partindo de 0 3: economicamente inviavel partindo de 0 |

Observações:

- Para a realização dos testes automáticos, a compilação se dará da seguinte forma:
g++ -pedantic -Wall *.cpp -lm -lutil