

Tarefa N° 03 - Destinos Viáveis

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

Linguagem para implementação: **C.**

Professor: Andrei Braga

Destinos Viáveis

Considere o seguinte problema. Uma empresa aérea está planejando uma reformulação da sua malha de voos. 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 existir da cidade de origem para a de destino uma sequência de voos cujo custo total seja menor ou igual a um valor máximo definido pela empresa.

Nesta tarefa, você deve implementar uma estrutura de dados para armazenar um grafo **dirigido** (digrafo) **simples** que possua **pesos não-negativos nas arestas**. O digrafo deve ser representado como **listas de adjacência**. Você deve escrever uma função 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 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 **V** ($V \geq 0$), sendo **C** o número de cidades atendidas na malha de voos da empresa e **V** o número de voos da malha. Cada uma das **V** linhas seguintes contém três inteiros **X**, **Y** e **Z**, indicando que a malha de voos da empresa contém um voo da cidade **X** para a cidade **Y** com custo **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 máximo de custo **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 o custo total mínimo de uma sequência de voos de **x** para **u** seja menor ou igual ao valor máximo definido,

- este custo seguido do caractere , e
- os índices das cidades de uma sequência de voos de **x** para **u** de custo total mínimo – os índices das cidades devem ser impressos antecidos por um espaço em branco e na ordem em que aparecem na sequência de voos **no sentido** de **u** para **x**;
- caso o custo total mínimo de uma sequência de voos de **x** para **u** seja maior que o valor máximo definido ou caso não exista uma sequência como esta,
 - o texto a seguir:
sem caminho viavel partindo de **x**
- Exceto para a primeira reavaliação a ser feita pela empresa, o seu programa deve imprimir uma linha em branco antes das impressões relativas à reavaliação.

Exemplo de execução 1

```

4 5
0 1 6
0 2 2
2 3 3
3 0 4
3 1 2
1
0 2
1: sem caminho viavel partindo de 0
2: 2, 2 0
3: sem caminho viavel partindo de 0

```

Exemplo de execução 2

```

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: 1, 1 2
3: sem caminho viavel partindo de 2

1: sem caminho viavel partindo de 0
2: sem caminho viavel partindo de 0
3: sem caminho viavel partindo de 0

```

Nota: Textos em azul denotam dados de entrada do programa.

Textos em vermelho denotam dados de saída do programa.

Observações:

- Para a realização dos testes automáticos, a compilação se dará da seguinte forma:
gcc -std=c99 -pedantic -Wall *.c -lm -lutil