

Tarefa Nº 02 - Alterações de Trânsito

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

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

Professor: Andrei Braga

Alterações de Trânsito

O prefeito de uma determinada cidade resolveu tentar melhorar o trânsito da cidade alterando o sentido de tráfego de várias de suas vias. Por falta de conhecimento do prefeito sobre grafos, não houve um planejamento adequado para as alterações. Como consequência, o trânsito ficou muito pior na cidade, deixando os seus moradores indignados. Resultado: o prefeito sofreu um *impeachment*.

A nova administração da cidade decidiu ser mais racional para tentar resolver o problema. O primeiro passo foi buscar entender melhor a situação atual do trânsito da cidade. Para isso, você foi contratado para realizar o seguinte:

1. Construir um grafo dirigido (digrafo) onde cada vértice representa um ponto do sistema de vias da cidade e cada aresta representa uma via com um sentido de um ponto para outro.
2. Usando o **algoritmo** visto na aula do dia **24/09/2024**, determinar as componentes fortemente conexas do grafo do sistema de vias da cidade.
3. Para cada componente fortemente conexa c_j do grafo do sistema de vias da cidade, calcular quantas outras componentes fortemente conexas c_j existem tais que há uma aresta vw de um vértice v que pertence a c_j para um vértice w que pertence a c_j .

Nesta tarefa, você deve implementar uma classe que representa um **digrafo simples**. Você deve escrever um método que executa no digrafo uma operação que realiza o que está descrito no Item 3 acima.

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 **P** ($P \geq 1$) e **D** ($D \geq 1$), sendo **P** o número de pontos do sistema de vias da cidade e **D** o número de descritores de vias que vêm a seguir. Cada uma das próximas **D** linhas contém três inteiros **X**, **Y** e **S**, indicando o seguinte:

- Se **S** é 1, então existe uma via do ponto **X** para o ponto **Y**;
- Se **S** é 2, então existe uma via do ponto **X** para o ponto **Y** e uma via do ponto **Y** para o ponto **X**.

Saída

Seja **C** o número de componentes fortemente conexas do grafo do sistema de vias da cidade. O seu programa deve considerar que os índices destas componentes são 0, 1, ..., **C** - 1. Além disso, se existe um vértice v que pertence a uma componente fortemente conexa c_i e um vértice w que pertence a uma componente fortemente conexa c_j tal que $v < w$, então o índice de c_i deve ser menor que o índice de c_j .

O seu programa deve imprimir **C** linhas, uma para cada uma das componentes fortemente conexas 0, 1, ..., **C** - 1, em sequência. Cada uma destas linhas deve conter o índice da componente seguido do caractere :, de um espaço em branco e do valor calculado no Item 3 acima.

Exemplos de execução

Entrada	Saída
3 2 0 1 2 0 2 1	0: 1 1: 0

Entrada	Saída
4 5 0 1 1 0 2 2 1 3 1 2 3 1 3 0 2	0: 0

Observações:

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