

Problema 2

Bloqueio Policial

Tempo limite: 0,25s (C/C++), 1,0s(Java)

Memória limite: 64 MB

Para procedimentos de segurança pública de qualquer cidade, é importante determinar trechos de ruas ou estradas que tenham maior chance de interceptar o tráfego de suspeitos ou criminosos dentro da cidade. Para o novo secretário de segurança nomeado pelo novo prefeito, os trechos de vias que devem ser escolhidos para bloqueios policiais devem ser aqueles que são a única forma de acesso a certos pontos a partir de algum outro ponto da cidade. Determinando esses trechos, ele acredita que ficará mais fácil e rápido de se cercar uma região onde um suspeito se localiza montando bloqueios policiais em trechos estratégicos, que são os acessos a essa região e que, em uma tentativa de fuga, o suspeito necessariamente teria que passar por algum desses trechos para sair dessa região.

A sua tarefa é, dado o mapa de uma cidade, determinar os trechos a serem considerados como possíveis locais para posicionar bloqueios policiais.

Entrada

Uma entrada para o programa apresentará um o mapa de uma cidade. A primeira linha contém dois inteiros N ($1 \leq N \leq 100.000$) e M ($0 \leq M \leq 100.000$), representando o número de pontos e o número de trechos conectando pares de pontos do mapa, respectivamente. Cada ponto é nomeado por um número distinto entre 0 e $N-1$. Nas próximas M linhas, são apresentados os pares de pontos a b que representam um trecho de via que liga o ponto a e b da cidade. Um trecho pode ser trafegado em ambas direções. Para cada par de pontos, existe no máximo um trecho e não há um trecho ligando um ponto a ele mesmo. Para cada ponto, é sempre possível encontrar um caminho a qualquer outro ponto da cidade.

Saída

Seu programa deve imprimir todos os trechos que podem receber bloqueios policiais. Imprima cada trecho em uma linha de forma ordenada. Cada trecho deve descrever o par de pontos que o trecho liga, sendo que aquele de menor nome (origem) deve ser impresso antes do de maior nome (destino). Trechos devem ser impressos em ordem ascendente dos pontos de origem. Para trechos com origens iguais, imprima na ordem ascendente de pontos de destino.

| Exemplo de Entrada | Exemplo de Saída |
|---------------------------------|------------------|
| 4 4 0 1 0 2 1 2 2 3 | 2 3 |

| Exemplo de Entrada | Exemplo de Saída |
|---|------------------|
| 13 17 0 1 1 2 2 3 3 4 4 1 0 5 0 12 5 12 5 6 5 7 7 8 7 9 8 9 9 10 9 11 9 12 10 11 | 0 1 5 6 |

