

# Escalonamento ótimo

Nome do arquivo fonte: `escalona.c`, `escalona.cpp`, `escalona.pas`, `escalona.java`, ou `escalona.py`

O SBC (*System for Batch Computing*) é um sistema operacional voltado para a execução sequencial de tarefas. O operador do sistema cria tarefas e o sistema operacional é responsável por agendar a execução destas tarefas.

Cada tarefa pode depender da conclusão de algumas tarefas para poder começar. Se uma tarefa  $A$  depende de uma tarefa  $B$ , a tarefa  $B$  deve terminar antes que a tarefa  $A$  inicie sua execução.

Além disto, cada tarefa possui uma prioridade. É sempre mais vantajoso para o sistema começar executando uma tarefa de mais alta prioridade, depois continuar executando uma tarefa de mais alta prioridade dentre as que sobraram e assim por diante.

Neste problema, será dado um inteiro  $N$ , que irá representar o número de tarefas no sistema. As tarefas serão numeradas de 0 até  $N - 1$ . Tarefas com índice menor possuem prioridade maior, de forma que a tarefa 0 é a tarefa de mais alta prioridade, a tarefa 1 é a tarefa com a segunda maior prioridade e assim por diante, até a tarefa  $N - 1$ , que é a tarefa com a menor prioridade. Além disso, serão dadas  $M$  relações de dependência entre as tarefas.

Seu objetivo será decidir se é possível executar as tarefas em alguma ordem. Caso seja possível, você deverá produzir uma ordem de execução ótima para as tarefas, isto é, desempate as ordens possíveis pela prioridade da primeira tarefa. Se o empate ainda persistir, desempate pela prioridade da segunda tarefa, e assim por diante.

## Entrada

A primeira linha da entrada contém inteiros  $N$  e  $M$ . As próximas  $M$  linhas descrevem, cada uma, uma dependência entre as tarefas da entrada. Cada uma dessas linhas irá conter dois inteiros  $A$  e  $B$  que indicam que a tarefa  $B$  depende da tarefa  $A$ , isto é, que a tarefa  $A$  deve terminar antes que a tarefa  $B$  inicie.

## Saída

Se não for possível ordenar as tarefas de forma que as dependências sejam satisfeitas, imprima uma única linha contendo o caracter “\*”. Caso contrário, imprima  $N$  linhas contendo cada uma um número inteiro. O inteiro na  $i$ -ésima linha deve ser o índice da  $i$ -ésima tarefa a ser executada na ordem ótima de execução das tarefas.

## Restrições

- $0 \leq N \leq 50000$ .
- $0 \leq M \leq 200000$ .
- $0 \leq A, B < N$ .

## Informações sobre a pontuação

- Em um conjunto de casos de teste totalizando 60 pontos,  $N \leq 1000$ .



Exemplos

Entrada	Saída
3 1	1
2 0	2
	0

Entrada	Saída
2 2	*
0 1	
1 0	