

# Chi si ferma è perduto (dontstop)

## Descrizione del problema

Due giocatori si alternano nello spostare una pedina da un nodo ad un altro di un grafo diretto. Nei turni pari (dispari), il giocatore  $P$  (il giocatore  $D$ ) guarda in quale nodo del grafo si trovi la pedina, e guarda quali archi escano da quel nodo ed in quali nodi tali archi portino. Se non esce alcun arco, allora il giocatore di turno non può muovere la pedina e perde, altrimenti sceglie lungo quale arco uscente muovere la pedina e riconsegna all'avversario la situazione di gioco con la pedina collocata nel nodo verso cui porta l'arco scelto. La cosa che nessuno dei due vuole nel modo più assoluto è perdere, dopodichè entrambi i giocatori vorrebbero far perdere l'avversario, cosa che considerano vittoria. Il gioco può proseguire indefinitivamente, ma, se un giocatore riesce a far perdere l'avversario è questa la direzione in cui andrà a remare.

Scrivi una procedura che stabilisca, per ogni possibile nodo di partenza per la pedina, l'esito della partita ove entrambi i giocatori giochino ottimamente. I nodi sono numerati da 0 a  $N - 1$ . Per ogni nodo  $u$ , il valore  $\text{esito}[u]$  denota l'esito di una partita, ottimamente giocata da entrambi i giocatori, che abbia preso avvio con la pedina collocata sul nodo  $u$ , e vale:

- +1 se il giocatore che muove per primo ha una strategia vincente,
- 1 se il giocatore che muove per secondo ha una strategia vincente,
- 0 se entrambi i giocatori hanno una strategia che consente loro di non perdere mai.

Dovrai calcolare e stampare  $\text{esito}[u]$  per ciascun nodo  $u = 0, \dots, N - 1$ .

## Assunzioni

- $1 \leq N \leq 100\,000$ ,  $1 \leq M \leq 1\,000\,000$ .
- non vi sono archi ripetuti (ma è possibile avere archi in entrambe le direzioni fra due nodi).

## Dati di input

Ogni riga del file `input.txt` contiene due numeri separati da spazio. L'ordine tra i due numeri è importante: nella prima riga troviamo prima  $N$  e poi  $M$ , e le successive  $M$  righe contengono gli archi del grafo diretto. Nello specifico, la riga  $1 + i$ , per  $i = 0, \dots, M - 1$ , contiene due numeri  $u_i$  e  $v_i$  che denotano un arco da  $u_i$  a  $v_i$ .

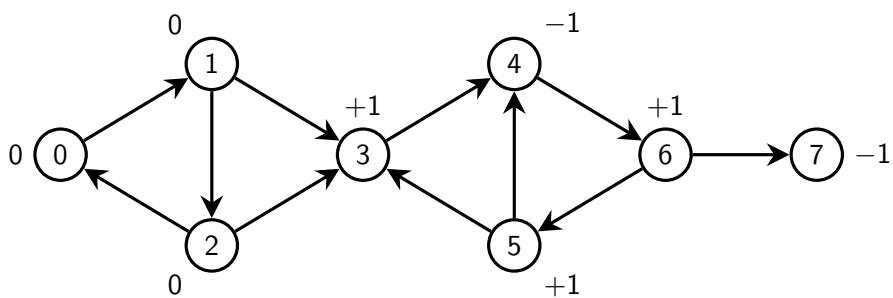
## Dati di output

Nel file `output.txt` si scriva un'unica riga contenente gli  $N$  numeri  $\text{esito}[u]$ , per  $u = 0, \dots, N - 1$ , così ordinati sull'indice  $u$  e separati da spazio.

## Esempio di input/output

File input.txt	File output.txt
<pre> 8 11 0 1 2 0 1 2 1 3 2 3 3 4 5 3 5 4 4 6 6 5 6 7 </pre>	<pre> 0 0 0 1 -1 1 1 -1 </pre>

Il file di esempio fa riferimento al seguente grafo. Il valore  $esito[u]$  è riportato in prossimità di ciascun nodo  $u$ .



## Subtask

- **Subtask 0 [ 0 punti]:** casi di esempio.
- **Subtask 1 [20 punti]:**  $N \leq 10$ , il grafo è aciclico.
- **Subtask 2 [20 punti]:**  $N \leq 10$ .
- **Subtask 2 [30 punti]:**  $N \leq 1\,000$ .
- **Subtask 2 [30 punti]:** nessuna limitazione specifica.