

## World directors (directors)

Linus in Tinus sta bila nedavno imenovana za nova *Direktorja sveta*.

Svet sestavlja  $N$  mest, oštevilčenih od 0 do  $N - 1$ , in  $N - 1$  dvosmernih cest. Med vsakim parom mest je mogoče potovati z nekim zaporedjem cest. Razdalja med dvema mestoma je najmanjše število cest, ki jih je treba prečkati, da se premaknemo iz enega mesta v drugo.

Kot novoimenovana direktorja morata Linus in Tinus opraviti tradicionalen nadzor sveta, ki obsega naslednje:

- Najprej se oba postavita na začetni mesti  $X$  in  $Y$ .
- Nato se vsak dan eden od njiju premakne iz svojega trenutnega mesta v neko sosednje. To se ponavlja, dokler ne obiščeta vsakega mesta vsaj enkrat in oba stojita na svojih začetnih položajih. Upoštevajte, da se jima ni treba izmenično premikati, lahko se npr. Linus premika več dni zapored.

Linus in Tinus zelo dobro vesta, da bo njuna patrulja veljala za bolj slovesno, čim bolj oddaljena bosta drug od drugega: *slovesnost* patrulje je razdalja med njima, ko sta si najbližje.

Linus in Tinus sta vas najela zato, da jima pomagata načrtovati njihovo patruljo, in vaša naloga je odgovoriti na  $Q$  vprašanj naslednje oblike:

- Če Linus začne v mestu  $X$  in Tinus v mestu  $Y$ , kakšna je največja slovesnost, ki jo lahko dosežeta z njuno patruljo?

## Implementacija

Vaša rešitev mora vsebovati eno datoteko z končnico `.cpp`.

↩ Med prilogami k tej nalogi boste našli predlogo `directors.cpp` z vzorčno implementacijo.

Implementirati morate naslednjo funkcijo:

C++	<code>void init(int N, vector&lt;int&gt; A, vector&lt;int&gt; B);</code>
-----	--

- Celo število  $N$  je število mest.
- Polji  $A$  in  $B$ , ki ju indeksiramo od 0 do  $N - 2$ , vsebujeta ceste. Natančneje,  $i$ -ta cesta povezuje  $A_i$  in  $B_i$ .
- Funkcijo se pokliče enkrat, na začetku izvajanja vaše rešitve.

C++	<code>int patrol(int X, int Y);</code>
-----	--

- Celi števili  $X$  in  $Y$  sta začetni mesti Linusa in Tinusa.
- Funkcija naj vrne največjo slovesnost patrulje, kjer Linus in Tinus začneta v mestih  $X$  in  $Y$ .
- Funkcijo se pokliče  $Q$ -krat med izvajanjem vaše rešitve.

## Primer ocenjevalnika

Poenostavljena različica ocenjevalnika, ki se uporablja pri preverjanju, je na voljo v imeniku, povezanem s to nalogo. Uporabite jo lahko za lokalno testiranje svojih rešitev. Vzorčni ocenjevalnik prebere

vhodne podatke iz `stdin`, kliče funkcijo, ki jo morate implementirati, in zapiše v `stdout` v naslednji obliki.

Vhodna datoteka je sestavljena iz  $N + Q$  vrstic, ki vsebujejo:

- Vrstica 1: celi števili  $N$  in  $Q$ .
- Vrstica  $2 + i$  ( $0 \leq i < N - 1$ ): celi števili  $A_i$  in  $B_i$ .
- Vrstica  $N + 1 + j$  ( $0 \leq j < Q$ ): celi števili  $X_j$  in  $Y_j$ .

Izhodna datoteka je sestavljena iz  $Q$  vrstic, ki vsebujejo vrednosti, ki jih vrne funkcija `patrol`.

## Omejitve

- $1 \leq N \leq 200\,000$ .
- $1 \leq Q \leq 100\,000$ .
- $0 \leq X, Y < N$  pri vsakem poizvedbi.

## Točkovanje

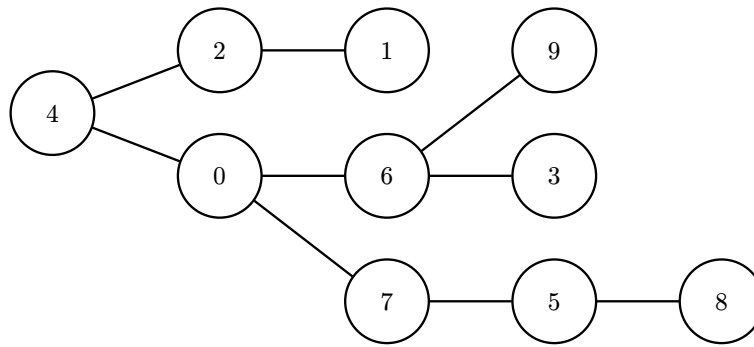
- **Podnaloga 0** [ 0 točk]: Primer.
- **Podnaloga 1** [ 8 točk]:  $A_i = 0$ ,  $B_i = i + 1$  za vse  $0 \leq i < N - 1$ .
- **Podnaloga 2** [16 točk]:  $A_i = 0$ ,  $B_i = i + 1$  ali  $A_i = i$ ,  $B_i = i + 1$  za vse  $0 \leq i < N - 1$ .
- **Podnaloga 3** [13 točk]:  $N, Q \leq 200$ .
- **Podnaloga 4** [14 točk]:  $N \leq 1000$ .
- **Podnaloga 5** [18 točk]: Pri vsakem poizvedbi  $X$  in  $Y$  maksimirajte odgovor glede na vsa začetna mesta.
- **Podnaloga 6** [17 točk]:  $Q \leq 200$ .
- **Podnaloga 7** [14 točk]: Brez dodatnih omejitev.

## Primeri vhoda/izhoda

stdin	stdout
10 3	2
0 4	1
1 2	2
8 5	
6 0	
9 6	
2 4	
7 0	
3 6	
5 7	
9 8	
0 6	
6 4	

## Razlaga

V prvem vzorčnem primeru ima svet naslednjo strukturo:



Za prvo poizvedbo lahko dosežemo slovesnost 2 z naslednjimi premiki:

- Linus potuje:  $9 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ;
- Tinus potuje:  $8 \rightarrow 5 \rightarrow 7 \rightarrow 0 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 9$ ;
- Linus potuje:  $1 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 7 \rightarrow 5 \rightarrow 8$ ;
- Tinus potuje:  $9 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ;
- Linus potuje:  $8 \rightarrow 5 \rightarrow 7 \rightarrow 0 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 9$ ;
- Tinus potuje:  $1 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 7 \rightarrow 5 \rightarrow 8$ .

Pokažemo lahko, da patrolja slovesnosti 3 ali več ne obstaja.