

## World directors (directors)

Linus și Tinus au fost numiți recent noii doi *Directori ai Lumii*.

Lumea e formată din  $N$  orașe, numerotate de la 0 la  $N - 1$ , și  $N - 1$  drumuri bidirecționale. Se poate ajunge între orice pereche de orașe folosind o secvență de drumuri. Distanța dintre două orașe e numărul minim de drumuri care trebuie parcurse pentru a te deplasa de la un oraș la celălalt.

Ca directori nou numiți, Linus și Tinus trebuie să efectueze o *patrulare* tradițională a întregii lumi, care constă în următoarele:

- Mai întâi, Linus și Tinus se deplasează în niște orașe inițiale  $X$  și  $Y$ .
- Apoi, în fiecare zi, unul dintre cei doi directori se mută din orașul curent într-unul accesibil printr-un singur drum de la el. Asta continuă până când fiecare director a fost în fiecare oraș cel puțin o dată și s-au întors în orașele lor inițiale. Rețineți că un director se poate mișca mai multe zile la rând: cei doi directori nu trebuie să alterneze.

Linus și Tinus știu foarte bine că patrularea lor va fi considerată mai solemnă cu cât sunt mai distanțați unul de celălalt: *solemnitatea* unei patrulări e distanța dintre cei doi directori atunci când sunt cel mai aproape unul de celălalt.

Cei doi directori te-au angajat, așadar, să-i ajuți să-și planifice patrularea și treaba ta e să răspunzi la  $Q$  întrebări de tipul următor:

- Dacă Linus pornește din orașul  $X$  și Tinus din orașul  $Y$ , care e solemnitatea maximă pe care o pot atinge cu patrularea lor?

## Implementare

Trebuie să trimiți un singur fișier cu extensia `.cpp`.



Printre atașamentele acestei probleme, vei găsi un șablon `directors.cpp` cu o implementare exemplu.

Trebuie să implementezi următoarele funcții:

C++

```
void init(int N, vector<int> A, vector<int> B);
```

- Întregul  $N$  e numărul de orașe.
- Array-urile  $A$  și  $B$ , indexate de la 0 la  $N - 2$ , conțin drumurile. În particular, al  $i$ -lea drum conectează  $A_i$  și  $B_i$ .
- Funcția va fi apelată o singură dată la începutul execuției programului tău.

C++

```
int patrol(int X, int Y);
```

- Întregii  $X$  și  $Y$  sunt orașele de pornire ale lui Linus și Tinus.
- Funcția ar trebui să returneze solemnitatea maximă a unei patrulări în care Linus și Tinus pornesc din orașele  $X$  și  $Y$ .
- Funcția va fi apelată de  $Q$  ori pe parcursul execuției programului tău.

## Grader de probă

O versiune simplificată a graderului folosit în timpul corectării e disponibilă în directorul aferent acestei probleme. O poți folosi pentru a-ți testa soluțiile locale. Sample grader-ul citește datele de intrare din `stdin`, apelează funcția pe care trebuie să o implementezi și scrie în `stdout` în următorul format.

Fișierul de intrare e compus din  $N + Q$  linii, conținând:

- Linia 1: întregii  $N$  și  $Q$ .
- Linia  $2 + i$  ( $0 \leq i < N - 1$ ): întregii  $A_i$  și  $B_i$ .
- Linia  $N + 1 + j$  ( $0 \leq j < Q$ ): întregii  $X_j$  și  $Y_j$ .

Fișierul de ieșire e compus din  $Q$  linii, conținând valorile returnate de funcția `patrol`.

## Constrângeri

- $1 \leq N \leq 200\,000$ .
- $1 \leq Q \leq 100\,000$ .
- $0 \leq X, Y < N$  în fiecare interogare.

## Punctaj

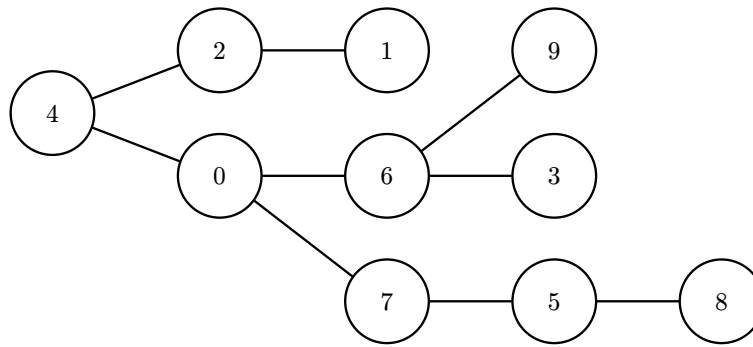
- **Subtask-ul 0 [ 0 puncte]:** Exemplu.
- **Subtask-ul 1 [ 8 puncte]:**  $A_i = 0$ ,  $B_i = i + 1$  pentru toți  $0 \leq i < N - 1$ .
- **Subtask-ul 2 [16 puncte]:**  $A_i = 0$ ,  $B_i = i + 1$  sau  $A_i = i$ ,  $B_i = i + 1$  pentru toți  $0 \leq i < N - 1$ .
- **Subtask-ul 3 [13 puncte]:**  $N, Q \leq 200$ .
- **Subtask-ul 4 [14 puncte]:**  $N \leq 1000$ .
- **Subtask-ul 5 [18 puncte]:** În fiecare interogare,  $X$  și  $Y$  maximizează răspunsul peste toate orașele de pornire.
- **Subtask-ul 6 [17 puncte]:**  $Q \leq 200$ .
- **Subtask-ul 7 [14 puncte]:** Fără constrângeri suplimentare.

## Exemple de intrare/ieșire

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	

## Explicație

În **primul exemplu** lumea are următoarea structură:



Pentru prima interogare putem obține o patrulare de solemnitate 2 cu următoarele mișcări:

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

Se poate demonstra că nu există nicio patrulare cu solemnitate 3 sau mai mare.