

World directors (directors)

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

Lumea este formată din N orașe și $N - 1$ drumuri, iar între orice pereche de orașe se poate circula folosind o secvență de drumuri. Distanța dintre două orașe este numărul minim de drumuri care trebuie parcurse pentru a te deplasa dintr-un oraș în celălalt.

În calitate de directori nou numiți, Linus și Tinus trebuie să efectueze o *patrulă* 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 deplasează din orașul curent într-un oraș accesibil printr-un singur drum. Acest lucru continuă până când ambii directori au fost în fiecare oraș cel puțin o dată și s-au întors în orașele lor inițiale. Rețineți că un director se poate deplasa mai multe zile la rând: cei doi directori nu trebuie să alterneze.

Linus și Tinus știu foarte bine că patrula lor va fi considerată mai solemnă cu cât sunt mai distanți unul de celălalt: *solemnitatea* unei patrule este 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 patrula, iar treaba ta este să răspunzi la Q întrebări de tipul următor:

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

Implementare

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



Printre atașamentele acestei sarcini, 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);
```

- Numărul întreg N reprezintă numărul de orașe.
- Vectorii A și B , indexați de la 0 la $N - 2$, conțin drumurile. Mai exact, 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);
```

- Numerele întregi X și Y sunt orașele de pornire ale lui Linus și Tinus.
- Funcția ar trebui să returneze solemnitatea maximă a unei patrule 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 la corectare este disponibilă în directorul aferent acestei probleme. O poți folosi pentru a-ți testa soluțiile locale. Graderul exemplu citește datele de intrare din `stdin`, apelează funcția pe care trebuie să o implementezi și scrie în `stdout` în următorul format.

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

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

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

Constrângeri

- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 100\,000$.
- Muchiile (A_i, B_i) descriu un arbore valid.
- $0 \leq X, Y < N$ în fiecare interogare.

Punctaj

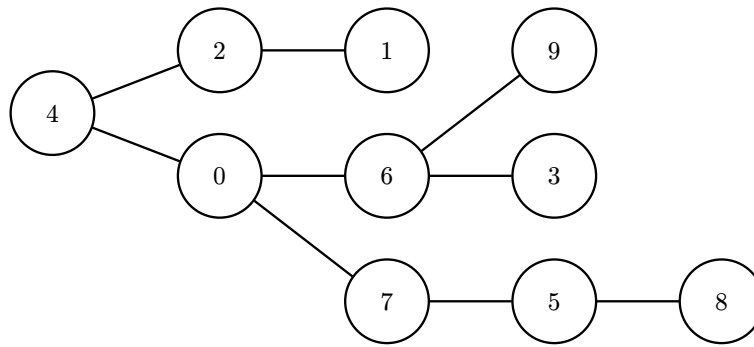
- **Subtask-ul 0 [0 puncte]:** Exemplu.
- **Subtask-ul 1 [8 puncte]:** $A_i = 0, B_i = i + 1$ pentru toți $0 \leq i < N - 1$, adică arborele este o stea.
- **Subtask-ul 2 [16 puncte]:** $A_i = 0, B_i = i + 1$ sau $A_i = i, B_i = i + 1$ pentru toți $0 \leq i < N - 1$.
- **Subtask-ul 3 [13 puncte]:** $N, Q \leq 200$.
- **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 avem următorul arbore:



Pentru prima interogare putem realiza o patrulă cu solemnitatea 2 prin următoarele mișcări:

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

Se poate demonstra că nu există o patrulă cu solemnitatea 3 sau mai mare.