Railway Schedule (paths)

The railway network in the *Pordenone* county consists of N train stations connected by $N-1$ tracks (X_i, Y_i) so that from every station is possible to reach any other station: in other words, the tracks form a *tree*.

This choice makes the transportation system extremely inefficient: trains going in opposite directions cannot cross each other on a single track, so they need to perform lengthy and complex manoeuvres to pass each other. The new administration founded its campaign trail on changing this situation once and for all... and now it's time to keep promises!

Figure 1: The Pordenone railway network.

Edoardo, the local leading expert in logistics, already has a mind-blowing idea for fixing the situation: making each track one-way, so that no crossings will ever occur! Of course, the tricky part is choosing the orientations so that the service remains acceptable for the majority of the population. After inspecting the traffic patterns, Edoardo discovered that most people travel between one of M pairs (A_i, B_i) of stations. Thus, an orientation of the tracks will be considered *acceptable* by the population only if for each such pair, either a path from A_i to B_i or a path from B_i to A_i should exist.

However, many acceptable orientations exist and Edoardo cannot choose among them, otherwise his system would be deemed as unfair: the only solution is to use *all* of them in a periodic schedule of daily track orientations. Help Edoardo design such a schedule by counting how many acceptable orientations exist! Since this number may be large, report it **modulo 1 000 000 007**.

✍ The *modulo* operation ($a \bmod m$) can be written in C/C++ as `(a % m)` and in Pascal as `(a mod m)`. To avoid the *integer overflow* error, remember to reduce all partial results through the modulus, and not just the final result!

☞ Among the attachments of this task you may find a template file `paths.*` with a sample incomplete implementation.

Input

The first line contains the only integer N. The following $N-1$ lines contain integers X_i, Y_i. The next line contains the only integer M. The last M lines contain integers A_i, B_i.

Output

You need to write a single line with an integer: the number of different acceptable orientations.

Constraints

- $1 \leq N, M \leq 300\,000$.
- $1 \leq X_i, Y_i, A_i, B_i \leq N$ for each $i = 0 \ldots N-1$.
- $X_i \neq Y_i$ and $A_i \neq B_i$ for each $i = 0 \ldots N-1$.

- The tracks connect all the stations together.
- There exists at least one acceptable orientation.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (15 points) $N, M \leq 100$ and $Y_i = X_i + 1$ (the tracks form a path).

– **Subtask 3** (20 points) $Y_i = X_i + 1$ (the tracks form a path).

– **Subtask 4** (10 points) $N, M \leq 10$.

– **Subtask 5** (25 points) $N, M \leq 1000$.

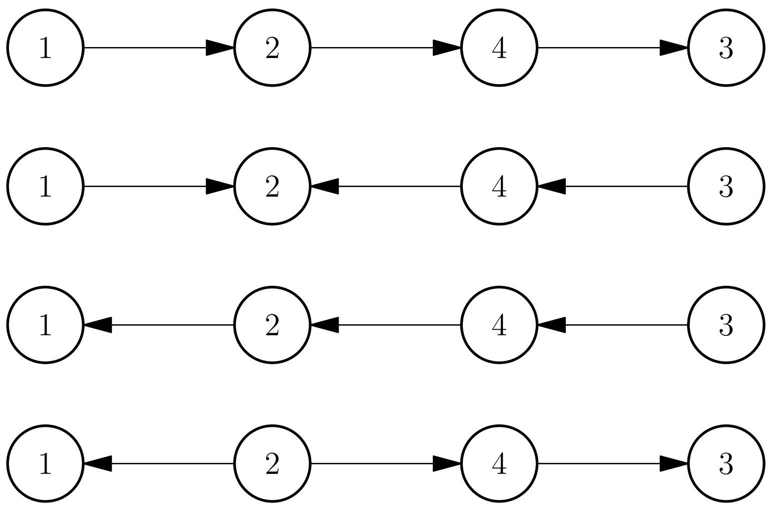
– **Subtask 6** (30 points) No additional limitations.

Examples

input	output
4	4
4 2	
3 4	
1 2	
1	
2 3	
6	2
1 3	
2 3	
3 4	
4 5	
4 6	
4	
1 6	
5 6	
3 4	
4 2	

Explanation

In the **first sample case**, the acceptable orientations are the following:



In the **second sample case**, the acceptable orientations are the following: