National Road System

There is a country with *n* cities. The cities connect with each other by the national road system which is in form of a tree. Therefore, the system contains *n-1* roads and there is always a path that connects two arbitrary cities. Because these roads are equivalent in term of length, the distance between two cities can be simplified as the number of roads in the path between these two cities.

Every city will have a tower that protects the city and its nearby cities from invaders. Each tower has a protecting range within which it can protect the surrounding cities. For example, a tower with the range of *d* can protect the cities whose distance from the tower is not more than *d*. Note that towers' protecting range is flexible and can be controlled by the government.

After running the towers for a long time, now the government has a plan to maintain these towers on the next *m* days. Only certain towers will operate on each day. According to the plan, on the *i-th* day, there will be only a_i towers that operate. The *j-th* tower of these ones will be located in the city c_j and have a protecting range of d_j. Now the government needs to know how many cities are protected by at least one tower on each day. Please help them.

Input

The first line contains the integer *n* ($1 \le n \le 50000$) – the number of cities.

Next, there are *n-1* following lines. Each line contains 2 integers *x*, *y* denoting there is a road between the city *x* and *y*.

The next line contains an integer *m* ($1 \le m \le 50000$) – the number of days in plan.

Then *m* lines follows. The *i-th* line contains an integer a_i– the number of running towers on *i-th* day, followed by a_i pairs of integers. The *j-th* pair consists of two integers c_j and d_j denoting the *j-th* tower's city and protecting range.

Note: The sum of a_i ($1 \le i \le$ m) is less than 500000.

Output

Output should contain *m* lines, each of which answers the number of cities that are protected on the respective day.

Examples

Standard Input	Standard Output

8
1 2
1 3
3 6
2 4
2 5
4 7
4 8
3
1 4 2
2 4 1 2 2
3 4 1 3 2 6 2
	6
	7
	7