

Winner Takes All

Translated by Gemini 3

Background

During the climax of UOJ Long Round #3, the core of the Flea Kingdom's internet infrastructure - the **CloudFlea** service - suddenly exploded! The blast stemmed from an erroneous configuration push, causing the CloudFlea routing table database to crash and severing global network services.

The contest website became inaccessible, the judge stopped responding, and the fleas were full of complaints. Emperor `__int128` urgently summoned the contestants and announced that this crisis would be turned into part of the competition: "Whoever can help fix CloudFlea will gain an advantage in the contest and hopefully inherit my title!"

Investigations revealed that CloudFlea's routing table uses a **01-Trie** to store routing rules for all IP addresses and requires fast querying capabilities.

Now, you need to help Emperor `__int128` solve the following problem to ensure the competition proceeds smoothly.

Problem Description

The routing table of CloudFlea is stored in a database as a **01-Trie** with n nodes (rooted at **1**).

For any node i in the tree, define s_i as the string obtained by concatenating the characters on the edges from the root to i in order (from shallow to deep). Note that $s_1 = \emptyset$ (the empty string).

There are q queries. Each query provides a node u and a set of (at most **3**) matching rules (a_i, c_i) . You need to count the number of nodes x that satisfy the following conditions:

1. s_x is a **suffix** of s_u (they can be identical, and s_x can be the empty string).
2. For all given rules (a_i, c_i) , if $a_i \leq |s_x|$, then the a_i -th character of s_x must be c_i ; otherwise, there is no restriction for that rule.

Your task is to answer all queries to help restore the CloudFlea routing service.

Input Format

The first line contains an integer n , representing the number of nodes in the 01-Trie.

The following $n - 1$ lines describe the tree structure. The i -th line (among these $n - 1$ lines) contains two integers f and c , indicating that the parent of node $i + 1$ is f , and the edge weight $(f, i + 1)$ is c .

The next line contains an integer q , representing the number of queries.

The following q lines describe the queries. Each line starts with two integers u and k , representing the query node and the number of constraints. This is followed by k pairs of integers $a_1, c_1, a_2, c_2, \dots, a_k, c_k$ representing the constraints.

It is guaranteed that a_i are strictly increasing within a query.

Output Format

For each query, output a single line containing one integer: the number of nodes x satisfying the conditions.

Examples

Example 1

Input

```
5
1 0
2 0
2 1
3 1
2
5 1 1 0
5 1 2 0
```

Output

```
3
2
```

Data Constraints

For 100% of the data, $2 \leq n, q \leq 2 \times 10^5$, $f_i < i$, $w_i, c_i \in \{0, 1\}$, and $1 \leq k \leq 3$.

Subtask	Score	$n, q \leq$	Special Properties	Dependencies
1	10	1000	None	None
2	20	2×10^5	w_i is generated randomly	None
3	20	2×10^5	$f_i = i - 1$ (Chain)	None
4	50	2×10^5	None	1, 2, 3

Time Limit: 4s Space Limit: 2GiB