

Judgement

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

You were preparing for the KAIST RUN ICPC 2025 Contest and completely forgot to submit your calculus homework. Now the professor has declared Judgement Day.

He has unleashed the legendary Hubo robot, programmed with only one mission: find you anywhere on campus and gently encourage you to finish the assignment.

The KAIST campus can be modeled as n buildings connected by $n - 1$ roads, forming a tree. Each road (u, v) has a **road congestion** $w(u, v)$.

When Hubo is at place x , it picks its next move randomly. For each neighbor y of x , the probability that Hubo moves from x to y is

$$\Pr[x \rightarrow y] = \frac{\frac{1}{w(x,y)}}{\sum_{z \sim x} \frac{1}{w(x,z)}},$$

where $w(x, y)$ is the congestion of road (x, y) and $z \sim x$ denotes all neighbors of x .

Unfortunately for you, the professor can alter the campus roads over time, changing their loads.

You must process q events:

- **1 id new_w** — the congestion of the road with index **id** becomes **new_w**;
- **2 u v** — you are hiding at place v , and Hubo is deployed at place u . Output the **expected number of steps** until Hubo reaches you. If $u = v$, the answer is 0.

Since the expectation can be rational, output it modulo $10^9 + 7$. Formally, if the expectation equals s/t in lowest terms, print $s \times t^{-1} \pmod{10^9 + 7}$, where t^{-1} is the modular inverse of t modulo $10^9 + 7$.

Input

The first line contains two integers n and q . Each of the next $n - 1$ lines contains three integers a, b, w describing a road between places a and b with congestion w ($1 \leq a, b \leq n$). Roads are indexed $1, \dots, n - 1$ in the order they appear. Each of the following q lines is one event in the formats above.

Output

For each query of type 2, print a single integer — the expected number of steps modulo $10^9 + 7$.

Scoring

- $3 \leq n, q \leq 2 \cdot 10^5$
- For every road congestion w , $1 \leq w \leq 10^9$

Example

standard input	standard output
4 7	9
1 2 1	9
2 3 1	10
3 4 1	22
2 1 4	0
2 4 1	
1 2 2	
2 1 4	
1 1 5	
2 4 1	
2 3 3	