Number of Ways to Arrive at Destination

You are in a city that consists of n intersections numbered from 0 to n - 1 with **bi-directional** roads between some intersections. The inputs are generated such that you can reach any intersection from any other intersection and that there is at most one road between any two intersections.

You are given an integer n and a 2D integer array roads where roads[i] = [u_i, v_i, $time_i$] means that there is a road between intersections u_i and v_i that takes $time_i$ minutes to travel. You want to know in how many ways you can travel from intersection 0 to intersection n - 1 in the **shortest amount of time**.

Return *the* ***number of ways*** *you can arrive at your destination in the* ***shortest amount of time***. Since the answer may be large, return it **modulo** $10^9 + 7$.

Example 1:

```
Input:
n=7, m=10
edges=
[[0,6,7],[0,1,2],[1,2,3],[1,3,3],[6,3,3],[3,5,1],[6,5,1],
[2,5,1],[0,4,5],[4,6,2]]

Output:
4
Explaination:

The four ways to get there in 7 minutes are:
- 0  6
- 0  4  6
- 0  1  2  5  6
- 0  1  3  5  6
```

Example 2:

```
Input:
n=6, m=8
edges=
[[0,5,8],[0,2,2],[0,1,1],[1,3,3],[1,2,3],[2,5,6],[3,4,2],
[4,5,2]]

Output:
3
Explaination:

The three ways to get there in 8 minutes are:
- 0  5
- 0  2  5
- 0  1  3  4  5
```

Constraints:

1 <= n <= 200

n - 1 <= roads.length <= n * (n - 1) / 2

roads[i].length == 3

0 <= u_i, v_i <= n - 1

1 <= $time_i$ <= 10^9

u_i != v_i

There is at most one road connecting any two intersections.
You can reach any intersection from any other intersection.

Expected Time Complexity: O(M * logN + N)
Expected Space Complexity: O(M+N)