Problem 1976: Number of Ways to Arrive at Destination

Problem Information

Difficulty: Medium
Acceptance Rate: 37.25%
Paid Only: No
Tags: Dynamic Programming, Graph, Topological Sort, Shortest Path

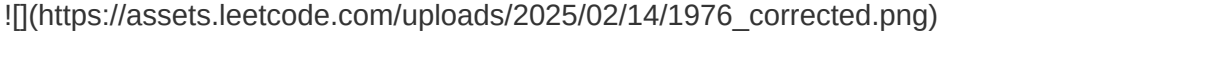
Problem Description

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] = [ui, vi, timei]` means that there is a road between intersections `ui` and `vi` that takes `timei` 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** `109 + 7`.

Example 1:



Input: n = 7, roads = [[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 **Explanation:** The shortest amount of time it takes to go from intersection 0 to intersection 6 is 7 minutes. 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 = 2, roads = [[1,0,10]] **Output:** 1 **Explanation:** There is only one way to go from intersection 0 to intersection 1, and it takes 10 minutes.

Constraints:

* `1 <= n <= 200` * `n - 1 <= roads.length <= n * (n - 1) / 2` * `roads[i].length == 3` * `0 <= ui, vi <= n - 1` * `1 <= timei <= 109` * `ui != vi` * There is at most one road connecting any two intersections. * You can reach any intersection from any other intersection.

Code Snippets

C++:

```
class Solution {
public:
int countPaths(int n, vector<vector<int>>& roads) {

}
};
```

Java:

```
class Solution {
public int countPaths(int n, int[][] roads) {

}
}
```

Python3:

```
class Solution:
def countPaths(self, n: int, roads: List[List[int]]) -> int:
```