

Problem 2959: Number of Possible Sets of Closing Branches

Problem Information

Difficulty: Hard

Acceptance Rate: 50.06%

Paid Only: No

Tags: Bit Manipulation, Graph, Heap (Priority Queue), Enumeration, Shortest Path

Problem Description

There is a company with n branches across the country, some of which are connected by roads. Initially, all branches are reachable from each other by traveling some roads.

The company has realized that they are spending an excessive amount of time traveling between their branches. As a result, they have decided to close down some of these branches (**possibly none**). However, they want to ensure that the remaining branches have a distance of at most maxDistance from each other.

The **distance** between two branches is the **minimum** total traveled length needed to reach one branch from another.

You are given integers n , maxDistance , and a **0-indexed** 2D array `roads`, where `roads[i] = [ui, vi, wi]` represents the **undirected** road between branches `ui` and `vi` with length `wi`.

Return `_` the number of possible sets of closing branches, so that any branch has a distance of at most maxDistance from any other `_`.

Note that, after closing a branch, the company will no longer have access to any roads connected to it.

Note that, multiple roads are allowed.

Example 1:

Input: $n = 3$, $\text{maxDistance} = 5$, $\text{roads} = [[0,1,2],[1,2,10],[0,2,10]]$ **Output:** 5

Explanation: The possible sets of closing branches are: - The set [2], after closing, active branches are [0,1] and they are reachable to each other within distance 2. - The set [0,1], after closing, the active branch is [2]. - The set [1,2], after closing, the active branch is [0]. - The set [0,2], after closing, the active branch is [1]. - The set [0,1,2], after closing, there are no active branches. It can be proven, that there are only 5 possible sets of closing branches.

Example 2:

Input: $n = 3$, $\text{maxDistance} = 5$, $\text{roads} = [[0,1,20],[0,1,10],[1,2,2],[0,2,2]]$ **Output:** 7

Explanation: The possible sets of closing branches are: - The set [], after closing, active branches are [0,1,2] and they are reachable to each other within distance 4. - The set [0], after closing, active branches are [1,2] and they are reachable to each other within distance 2. - The set [1], after closing, active branches are [0,2] and they are reachable to each other within distance 2. - The set [0,1], after closing, the active branch is [2]. - The set [1,2], after closing, the active branch is [0]. - The set [0,2], after closing, the active branch is [1]. - The set [0,1,2], after closing, there are no active branches. It can be proven, that there are only 7 possible sets of closing branches.

Example 3:

Input: $n = 1$, $\text{maxDistance} = 10$, $\text{roads} = []$ **Output:** 2 **Explanation:** The possible sets of closing branches are: - The set [], after closing, the active branch is [0]. - The set [0], after closing, there are no active branches. It can be proven, that there are only 2 possible sets of closing branches.

Constraints:

$1 \leq n \leq 10$ $1 \leq \text{maxDistance} \leq 105$ $0 \leq \text{roads.length} \leq 1000$ $0 \leq \text{roads}[i].\text{length} \leq 3$ $0 \leq u_i, v_i \leq n - 1$ $u_i \neq v_i$ $1 \leq w_i \leq 1000$ * All branches are reachable from each other by traveling some roads.

Code Snippets

C++:

```
class Solution {  
public:  
    int numberOfSets(int n, int maxDistance, vector<vector<int>>& roads) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int numberOfSets(int n, int maxDistance, int[][] roads) {  
  
    }  
}
```

Python3:

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