Problem 3067: Count Pairs of Connectable Servers in a Weighted Tree Network

Problem Information

Difficulty: Medium
Acceptance Rate: 55.13%
Paid Only: No
Tags: Array, Tree, Depth-First Search

Problem Description

You are given an unrooted weighted tree with `n` vertices representing servers numbered from `0` to `n - 1`, an array `edges` where `edges[i] = [ai, bi, weighti]` represents a bidirectional edge between vertices `ai` and `bi` of weight `weighti`. You are also given an integer `signalSpeed`.

Two servers `a` and `b` are **connectable** through a server `c` if:

* `a < b`, `a != c` and `b != c`. * The distance from `c` to `a` is divisible by `signalSpeed`. * The distance from `c` to `b` is divisible by `signalSpeed`. * The path from `c` to `b` and the path from `c` to `a` do not share any edges.

Return _an integer array_ `count` _of length_ `n` _where_ `count[i]` _is the**number** of server pairs that are **connectable** through_ _the server_ `i`.

Example 1:

![](https://assets.leetcode.com/uploads/2024/01/21/example22.png)

Input: edges = [[0,1,1],[1,2,5],[2,3,13],[3,4,9],[4,5,2]], signalSpeed = 1 **Output:** [0,4,6,6,4,0] **Explanation:** Since signalSpeed is 1, count[c] is equal to the number of pairs of paths that start at c and do not share any edges. In the case of the given path graph, count[c] is equal to the number of servers to the left of c multiplied by the servers to the right of c.

Example 2:

![](https://assets.leetcode.com/uploads/2024/01/21/example11.png)

Input: edges = [[0,6,3],[6,5,3],[0,3,1],[3,2,7],[3,1,6],[3,4,2]], signalSpeed = 3 **Output:** [2,0,0,0,0,0,2] **Explanation:** Through server 0, there are 2 pairs of connectable servers: (4, 5) and (4, 6). Through server 6, there are 2 pairs of connectable servers: (4, 5) and (0, 5). It can be shown that no two servers are connectable through servers other than 0 and 6.

Constraints:

* `2 <= n <= 1000` * `edges.length == n - 1` * `edges[i].length == 3` * `0 <= ai, bi < n` * `edges[i] = [ai, bi, weighti]` * `1 <= weighti <= 106` * `1 <= signalSpeed <= 106` * The input is generated such that `edges` represents a valid tree.

Code Snippets

C++:

```
class Solution {
public:
vector<int> countPairsOfConnectableServers(vector<vector<int>>& edges, int
signalSpeed) {

}
};
```

Java:

```
class Solution {
public int[] countPairsOfConnectableServers(int[][] edges, int signalSpeed) {

}
}
```

Python3:

```
class Solution:
def countPairsOfConnectableServers(self, edges: List[List[int]], signalSpeed:
int) -> List[int]:
```