# 1466. Reorder Routes to Make All Paths Lead to the City Zero

There are n cities numbered from 0 to n - 1 and n - 1 roads such that there is only one way to travel between two different cities (this network form a tree). Last year, The ministry of transport decided to orient the roads in one direction because they are too narrow.

Roads are represented by connections where connections[i] = [ai, bi] represents a road from city ai to city bi.

This year, there will be a big event in the capital (city 0), and many people want to travel to this city.

Your task consists of reorienting some roads such that each city can visit the city 0. Return the minimum number of edges changed.

It's guaranteed that each city can reach city 0 after reorder.

## SOLUTION IN JAVA

class Solution {

public int minReorder(int n, int[][] connections) {

List<Integer>[] graph = new List[n];

for (int i = 0; i < n; ++i)

graph[i] = new ArrayList<>();

for (int[] connection : connections) {

final int u = connection[0];

final int v = connection[1];

graph[u].add(v);

graph[v].add(-u); // - := u -> v

}

return dfs(graph, 0, -1);

}

private int dfs(List<Integer>[] graph, int u, int parent) {

int change = 0;

for (final int v : graph[u]) {

if (Math.abs(v) == parent)

continue;

if (v > 0)

++change;

change += dfs(graph, Math.abs(v), u);

}

return change;

}

}