# 547. Number of Provinces

There are n cities. Some of them are connected, while some are not. If city a is connected directly with city b, and city b is connected directly with city c, then city a is connected indirectly with city c.

A province is a group of directly or indirectly connected cities and no other cities outside of the group.

You are given an n x n matrix isConnected where isConnected[i][j] = 1 if the ith city and the jth city are directly connected, and isConnected[i][j] = 0 otherwise.

Return the total number of provinces.

## SOLUTION IN C++

class UnionFind {

public:

UnionFind(int n) : count(n), id(n), rank(n) {

iota(id.begin(), id.end(), 0);

}

void unionByRank(int u, int v) {

const int i = find(u);

const int j = find(v);

if (i == j)

return;

if (rank[i] < rank[j]) {

id[i] = j;

} else if (rank[i] > rank[j]) {

id[j] = i;

} else {

id[i] = j;

++rank[j];

}

--count;

}

int getCount() const {

return count;

}

private:

int count;

vector<int> id;

vector<int> rank;

int find(int u) {

return id[u] == u ? u : id[u] = find(id[u]);

}

};

class Solution {

public:

int findCircleNum(vector<vector<int>>& isConnected) {

const int n = isConnected.size();

UnionFind uf(n);

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

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

if (isConnected[i][j] == 1)

uf.unionByRank(i, j);

return uf.getCount();

}

};