1129. Shortest Path with Alternating Colors <medium>

class Solution {

public:

vector<int> shortestAlternatingPaths(int n, vector<vector<int>>& red\_edges, vector<vector<int>>& blue\_edges) {

vector<vector<pair<int,int>>> g(n); //index, {neighbor index, color}

for(auto& v:red\_edges)

g[v[0]].push\_back({v[1], 0});

for(auto& v:blue\_edges)

g[v[0]].push\_back({v[1], 1});

vector<vector<int>> vCost(n, vector<int>(2,-1));

queue<pair<int,int>> q; // index, color(0 or 1)

q.push({0,0});

q.push({0,1});

vCost[0] = {0,0};

while(!q.empty()){

auto [i, c1] = q.front(); q.pop();

for(auto [j, c2] : g[i]){

if(vCost[j][c2] != -1 || c1 == c2)

continue;

vCost[j][c2] = 1 + vCost[i][c1];

q.push({j, c2});

}

}

vector<int> res;

for(auto& v : vCost) {

sort(v.begin(), v.end());

res.push\_back(v[0] != -1 ? v[0] : v[1]);

}

return res;

}

};