**Worksheet-5**

**Student Name:-** Pushpraj Roy **UID:-** 20BCS9866

**Branch:-** BE- CSE **Section/Group:-** WM\_617 “A”

**Subjetct Code:-** 20CSP-314 **Semester:-** 5th

**Subject Name:-** Competitive Coding Lab

**Problem 1:-** Journey-to-the-moon

<https://www.hackerrank.com/challenges/journey-to-the-moon/problem?isFullScreen=true>

**Code:-**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

#include <map>

using namespace std;

vector<int> parent;

vector<int> rankk;

vector<int> v;

int find\_set (int v) {

if (v == parent[v])

return v;

return parent[v] = find\_set (parent[v]);

}

void union\_sets (int a, int b) {

a = find\_set (a);

b = find\_set (b);

if (a != b) {

if (rankk[a] < rankk[b])

swap (a, b);

parent[b] = a;

if (rankk[a] == rankk[b])

++rankk[a];

}

}

int n, m;

map<int,int> mm;

int main() {

cin >> n >> m;

parent.resize(n);

rankk.resize(n);

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

{

parent[i] = i;

rankk[i] = 0;

}

for (int i = 0; i != m; ++i)

{

int x,y;

cin >> x >> y;

union\_sets(x,y);

}

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

{

mm[find\_set(i)]++;

}

map<int,int>::iterator it = mm.begin();

map<int,int>::iterator itEnd = mm.end();

long long res = 0;

int b = 0;

for (; it != itEnd; ++it)

{

v.push\_back(it->second);

}

int l = v.size();

long long rr = 0;

for (int i = 0; i != l; ++i)

rr += v[i];

for (int i = 0; i != l; ++i)

{

rr -= v[i];

res += v[i]\*rr;

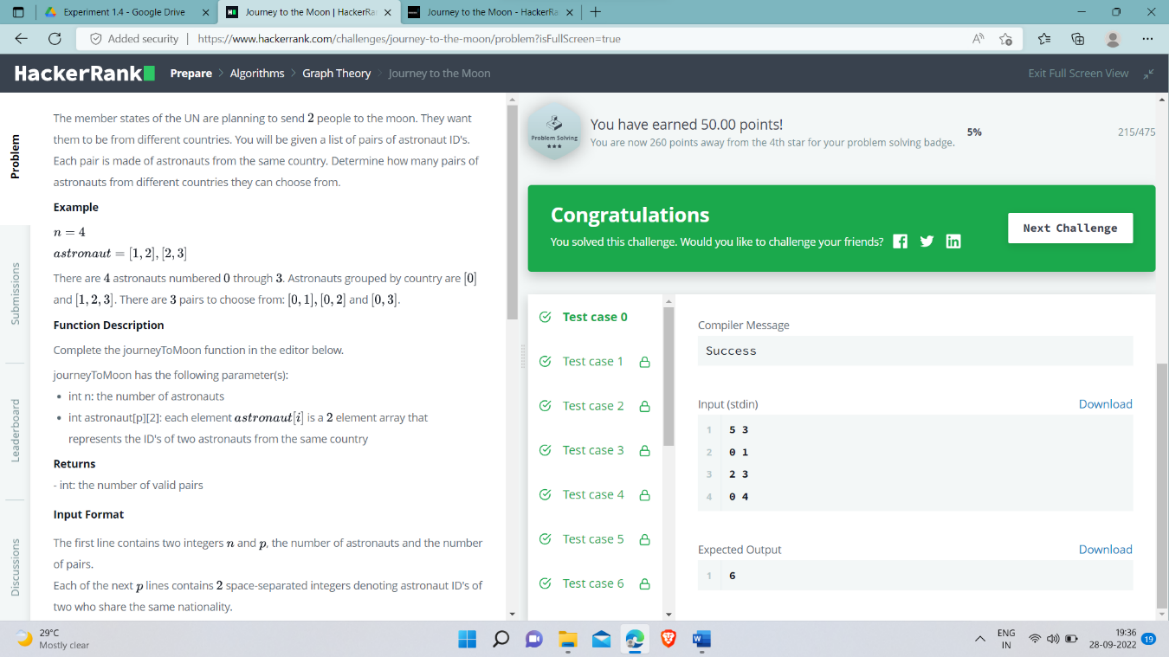
}

cout << res << endl;

return 0;

}

**Output:-**



**Problem 2:-** frog-in-maze

<https://www.hackerrank.com/challenges/frog-in-maze/problem?isFullScreen=true>

**Code:-**

#include <bits/stdc++.h>

#define endl '\n'

#define double long double

using namespace std;

const int MAXN = (42);

const double eps = 1e-12;

vector<double> gauss(vector<vector<double>> &a)

{

int n = a.size(), m = a[0].size() - 1;

vector<int> where(m, -1);

for(int col = 0, row = 0; col < m && row < n; col++)

{

int sel = row;

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

if(abs(a[i][col]) > abs(a[sel][col]))

sel = i;

if(abs(a[sel][col]) < eps) { where[col] = -1; continue; }

for(int i = col; i <= m; i++)

swap(a[sel][i], a[row][i]);

where[col] = row;

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

if(i != row)

{

if(abs(a[i][col]) < eps) continue;

double c = a[i][col] / a[row][col];

for(int j = 0; j <= m; j++)

a[i][j] -= c \* a[row][j];

}

row++;

}

vector<double> ans(m, 0);

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

if(where[i] != -1)

ans[i] = a[where[i]][m] / a[where[i]][i];

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

{

double sum = a[i][m];

for(int j = 0; j < m; j++)

sum -= ans[j] \* a[i][j];

if(abs(sum) > eps) return vector<double>();

}

return ans;

}

int n, m, k;

string a[MAXN];

int nxt\_x[MAXN][MAXN], nxt\_y[MAXN][MAXN];

void read()

{

cin >> n >> m >> k;

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

cin >> a[i];

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

for(int j = 0; j < m; j++)

nxt\_x[i][j] = i, nxt\_y[i][j] = j;

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

{

int x1, y1, x2, y2;

cin >> x1 >> y1 >> x2 >> y2;

x1--; y1--; x2--; y2--;

nxt\_x[x1][y1] = x2; nxt\_y[x1][y1] = y2;

nxt\_x[x2][y2] = x1; nxt\_y[x2][y2] = y1;

}

}

int N;

int encode(int x, int y) { return x \* m + y; }

int dirx[4] = {0, 0, 1, -1};

int diry[4] = {1, -1, 0, 0};

bool ok(int x, int y)

{

if(x >= n || y >= m || x < 0 || y < 0) return false;

return a[x][y] != '#';

}

void solve()

{

N = n \* m;

vector<vector<double> > matr;

vector<double> zero(N + 1, 0);

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

for(int j = 0; j < m; j++)

{

if(a[i][j] == '#') { matr.push\_back(zero); continue; }

else if(a[i][j] == '\*') { matr.push\_back(zero), matr[matr.size() - 1][encode(i, j)] = 1; continue; }

else if(a[i][j] == '%') { matr.push\_back(zero), matr[matr.size() - 1][encode(i, j)] = 1; matr[matr.size() - 1][N] = 1; continue; }

vector<int> adj;

for(int d = 0; d < 4; d++)

if(ok(i + dirx[d], j + diry[d]))

adj.push\_back(encode(nxt\_x[i + dirx[d]][j + diry[d]], nxt\_y[i + dirx[d]][j + diry[d]]));

matr.push\_back(zero);

matr[matr.size() - 1][encode(i, j)] = 1;

for(int v: adj)

matr[matr.size() - 1][v] = -((double)1 / (double)adj.size());

}

vector<double> ans = gauss(matr);

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

for(int j = 0; j < m; j++)

if(a[i][j] == 'A')

{

cout << setprecision(9) << fixed << ans[encode(i, j)] << endl;

return;

}

}

int main()

{

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

read();

solve();

return 0;

}

**Output:-**

