Min number of edjes to make all reachable from source in dg

int n, m, s;

vector<int> g1[MAXN], g2[MAXN], g3[MAXN];

int cnt[MAXN], comp[MAXN];

bool used[MAXN];

bool read() {

if (!(cin >> n >> m >> s))

return false;

forn (i, m) {

int u, v;

cin >> u >> v;

g1[u - 1].push\_back(v - 1);

g2[v - 1].push\_back(u - 1);

}

return true;

}

vector<int> order, vv;

void dfs1(int v) {

used[v] = true;

for (auto to : g1[v])

if (!used[to])

dfs1(to);

order.push\_back(v);

}

void dfs2(int v) {

vv.push\_back(v);

used[v] = true;

for (auto to : g2[v])

if (!used[to])

dfs2(to);

}

void dfs(int v) {

used[v] = true;

for (auto to : g3[v])

if (!used[to])

dfs(to);

}

void solve() {

--s;

forn (i, n)

if (!used[i])

dfs1(i);

fill(used, used + n, false);

reverse(order.begin(), order.end());

int cur = 0;

for (auto v : order)

if (!used[v]) {

vv.clear();

dfs2(v);

for (auto i : vv)

comp[i] = cur;

++cur;

}

s = comp[s];

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

for (auto to : g1[i])

if (comp[i] != comp[to]) {

g3[comp[i]].push\_back(comp[to]);

++cnt[comp[to]];

}

fill(used, used + cur, false);

dfs(s);

int ans = 0;

forn (i, cur)

if (cnt[i] == 0 && !used[i])

dfs(i), ++ans;

cout << ans << endl;

}

int main() {

cin.tie(nullptr);

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

std::mt19937 rand('S' + 'E' + 'R' + 'E' + 'Z' + 'H' + 'K' + 'A');

#ifdef SEREZHKA

freopen("file.in", "r", stdin);

#endif

while (read())

solve();

return 0;

};