// Hopcroft Karp Sample:

**#include <bits/stdc++.h>**

**#define MAXN 50005**

**#define INF 1000000007**

**#define tr(i,c) for(\_\_typeof((c).begin()) i=(c).begin();i!=(c).end();i++)**

**using namespace std;**

**int m, n, res = 0;**

**vector<int> g[MAXN];**

**int x[MAXN], y[MAXN], d[MAXN], q[MAXN];**

**bool FindPath() {**

**int L = 1, R = 0;**

**for (int u = 1; u <= n; u++)**

**if (x[u] == 0) {**

**d[u] = 0;**

**q[++R] = u;**

**} else**

**d[u] = INF;**

**d[0] = INF;**

**while (L <= R) {**

**int u = q[L++];**

**tr (i, g[u]) {**

**int v = \*i;**

**if (d[y[v]] == INF) {**

**d[y[v]] = d[u] + 1;**

**if (y[v])**

**q[++R] = y[v];**

**}**

**}**

**}**

**return (d[0] != INF);**

**}**

**bool dfs (int u) {**

**if (u == 0)**

**return (true);**

**tr (i, g[u]) {**

**int v = \*i;**

**if (d[y[v]] == d[u] + 1)**

**if (dfs (y[v])) {**

**x[u] = v;**

**y[v] = u;**

**return true;**

**}**

**}**

**d[u] = INF;**

**return false;**

**}**

**void GhepMax() {**

**while (FindPath()) {**

**for (int u = 1; u <= n; u++)**

**if (x[u] == 0)**

**if (dfs (u))**

**res++;**

**}**

**printf ("%d", res);**

**}**

**int main() {**

**if (fopen ("fmatch.inp", "r")) {**

**freopen ("fmatch.inp", "r", stdin);**

**freopen ("fmatch.out", "w", stdout);**

**}**

**int p;**

**scanf ("%d%d%d", &n, &m, &p);**

**for (int i = 1; i <= p; i++) {**

**int u, v;**

**scanf ("%d%d", &u, &v);**

**g[u].push\_back (v);**

**}**

**GhepMax();**

**}**