#include <stdio.h>

#include <string.h>

int e[10010][2];

int \*g[5010], \*num[5010], deg[5010], deg1[5010];

int s;

int t[5010];

bool v[10010], st, f;

void go(int p)

{

if (p == s && !st)

{

f = 1;

return;

}

int i;

st = 0;

for (i = deg[p]-1; i >= 0; i--)

if (!v[num[p][i]])

{

v[num[p][i]] = 1;

go(g[p][i]);

if (f)

{

t[p] = t[s];

return;

}

}

}

int main()

{

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

freopen("rpaths.out", "w", stdout);

int n, m, c, i, j;

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

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

{

scanf("%d%d", e[i], e[i]+1);

deg[--e[i][0]]++;

deg[--e[i][1]]++;

}

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

{

g[i] = new int[deg[i]+1];

num[i] = new int[deg[i]+1];

}

memset(deg, 0, sizeof(deg));

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

{

g[e[i][0]][deg[e[i][0]]] = e[i][1];

g[e[i][1]][deg[e[i][1]]] = e[i][0];

num[e[i][0]][deg[e[i][0]]++] = num[e[i][1]][deg[e[i][1]]++] = i;

}

c = 0;

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

{

if (t[i] == 0)

t[i] = ++c;

f = 1;

s = i;

memset(v, 0, sizeof(v));

while (f)

{

f = 0;

st = 1;

go(s);

}

}

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

for (j = deg[i]-1; j >= 0; j--)

if (t[i] != t[g[i][j]])

deg1[t[i]-1]++;

n = 0;

for (i = 0; i < c; i++)

n += (deg1[i] == 1);

printf("%d\n", (n+1)/2);

return 0;

}