#include <cstdio>

#include <vector>

#include <algorithm>

using namespace std;

#define MOD 1000000009

typedef vector<vector<long long> > VLL;

VLL s;

VLL mul(VLL a, VLL b)

{

long long n=a.size(), i, j, k;

VLL ret(n);

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

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

{

ret[i].push\_back(0);

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

ret[i][j] = (ret[i][j]+a[i][k]\*b[k][j])%MOD;

}

return ret;

}

VLL pow(long long p)

{

VLL r(s.size());

if (p == 0)

{

long long i, j;

for (i = 0; i < s.size(); i++)

for (j = 0; j < s.size(); j++)

r[i].push\_back(0);

for (i = 0; i < s.size(); i++)

r[i][i] = 1;

return r;

}

r = pow(p>>1);

if (p&1)

return mul(mul(r, r), s);

return mul(r, r);

}

int main()

{

VLL g;

long long n, m, d, k, l, i, j, p, q;

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

d--;

g.resize(n);

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

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

g[i].push\_back(0);

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

{

scanf("%lld%lld", &p, &q);

g[p-1][q-1] = g[q-1][p-1] = 1;

}

s = g;

s = pow(d);

p = m = 0;

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

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

m = (m+s[i][j])%MOD;

for (i = 1; i < (1<<k); i++)

{

l = 0;

s = g;

for (j = k-1; j >= 0; j--)

if ((i>>j)&1)

{

for (q = 0; q < s.size(); q++)

s[q].erase(s[q].begin()+j);

s.erase(s.begin()+j);

l++;

}

s = pow(d);

if (l&1)

{

for (q = 0; q < n-l; q++)

for (j = 0; j < n-l; j++)

p = (p-s[q][j]+MOD)%MOD;

p = (p+m)%MOD;

}

else

{

for (q = 0; q < n-l; q++)

for (j = 0; j < n-l; j++)

p = (p+s[q][j]+MOD)%MOD;

p = (p-m+MOD)%MOD;

}

}

printf("%lld\n", p);

return 0;

}