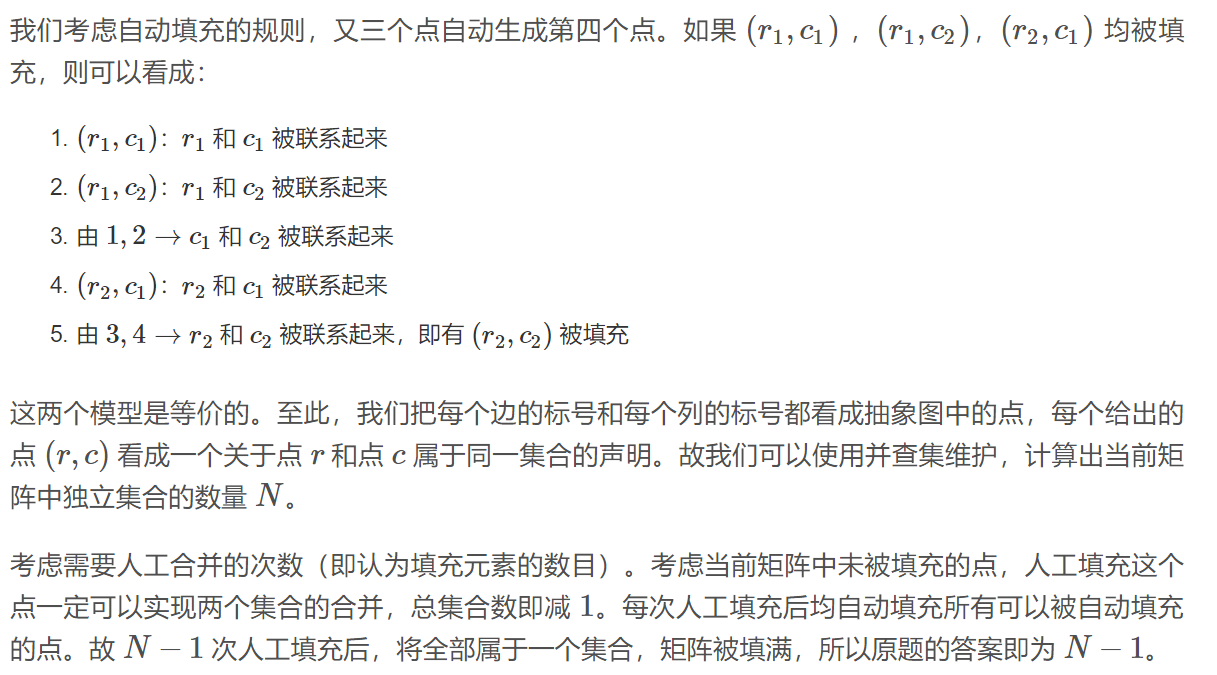


题解：



#include <cstdio>

#include <cstring>

#include<iostream>

#include<algorithm>

using namespace std;

typedef long long ll;

int f[1000000],depth[1000000];

bool b[1000000];

int Find(int x)

{

if(f[x]==x)return x;

return f[x]=Find(f[x]);//是否需要带路径压缩还要考虑一下

}

bool same(int x,int y)

{

return Find(x)==Find(y);

}

void Union(int x,int y)

{

x=Find(x);

y=Find(y);

if(x!=y)

{

if(depth[x]<depth[y])//按秩合并

{

f[x]=y;

}

else

{

f[y]=x;

if(depth[x]==depth[y])

depth[x]++;

}

}

}

int main()

{

int n,m,q,x,y;

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

memset(f,0,sizeof(f));

for(int i=1;i<=n+m;i++)

{

f[i]=i;

depth[i]=1;

}

while(q--)

{

scanf("%d%d",&x,&y);

Union(x,y+n);

}

ll ans(0);

for(int i=1;i<=n+m;i++)

{

int root=Find(i);

if(b[root]==0)

{

b[root]=1;

ans++;

}

}

cout<<ans-1<<endl;

return 0;

}