

## C 题题解

### 【描述】

在一个  $n*m$  的矩阵上有一些 hole，用  $1*2$  的积木去覆盖这些格子，hole 上不能覆盖，问能否覆盖所有的非 hole 格子。

### 【解法】

讲整张图黑白染色，以  $i+j$  为奇数的和为偶数的划分，如果两种类型格子数目不等，显然不存在解。从奇数点连边向偶数的点（这些点都不能是 hole），然后进行一次二分图匹配。如果最大匹配数等于格子数，那么就是可以完全覆盖的，否则无解。

### 【代码】

```
# include <cstring>
# include <cstdio>
# include <iostream>
# include <cstdlib>
using namespace std;
# define N 55
int mark[N][N], num[N][N];
int map[1000][10];
int vis[1000], result[1000];
int find (int x) //匹配过程
{
    if (vis[x])
        return 0;
    int v;
    vis[ x ]=1;
    for (int i=1; i<=map[x][0];i++)
    {
        v=map[x][i];
        if (result[v]==0 || find(result[v]))
        {
            result[v]=x;
            return 1;
        }
    }
    return 0;
}
int main (void)
{
    int n,m,k,x,y;
    cin>>n>>m>>k;
    for (int i=1;i<=k;i++)
    {
        cin>>y>>x;
```

```

        mark[x][y]=1;
    }
    int cnt1=0,cnt2=0;
    for (int i=1;i<=n;i++)
        for (int j=1;j<=m;j++)
            if (mark[i][j]==0)
            {
                if ((i+j)%2)
                    num[i][j]=++cnt1;
                else num[i][j]=++cnt2;
            }
    if (cnt1!=cnt2) //格子数不等无解
    {
        cout<<"NO"<<endl;
        return 0;
    }
    for (int i=1;i<=n;i++) //建图
        for (int j=1;j<=m;j++)
            if (num[i][j] && ((i+j)%2)) //只从奇点向偶点连边，这个点不能是 hole，orz，wa 了无数次
            {
                if (num[i-1][j])
                    map[num[i][j]][++map[num[i][j]][0]]=num[i-1][j];
                if (num[i+1][j])
                    map[num[i][j]][++map[num[i][j]][0]]=num[i+1][j];
                if (num[i][j-1])
                    map[num[i][j]][++map[num[i][j]][0]]=num[i][j-1];
                if (num[i][j+1])
                    map[num[i][j]][++map[num[i][j]][0]]=num[i][j+1];
            }
    int ans=0;
    for (int i=1 ; i<=cnt1; i++)
    {
        memset(vis,0,sizeof(vis));
        ans+=find(i);
    }
    if (ans>=cnt1)
        cout<<"YES"<<endl;
    else cout<<"NO"<<endl;
    return 0;
}

```