

中国象棋

一. 考察内容:

动态规划

二. 题目分析:

[题目大意]

在 $n*m$ 的棋盘放入一些“炮”，按照中国象棋的规则，使得其任何一个棋子不能被另外一个棋子一步吃掉，求出总方案数。

[写题思路]

这是一道神奇的动态规划题，和往常的动态规划题不同的是，该题的状态设定不是具体的标注每个棋子的摆放位置，而是记录模糊的位置关系，并通过乘法原理和组合数进行转移。

设状态 $f[i][j][k]$ 为前 i 行，其中 j 列放了0个棋子， k 列放了1个棋子的方案数，转移：

对于第 i 行，可以考虑不放任何棋子： $f[i][j][k]=f[i-1][j][k]$ ，

也可以放一个棋子，有两种方案，

1. 放到之前没放过棋子的一列中： $f[i][j][k]+=f[i-1][j+1][k-1]$

2. 放到之前放过一个棋子的一列中： $f[i][j][k]+=f[i-1][j][k+1]$

还可以放两个棋子，有三种方案，

1. 两个都放到之前没放过棋子的一列中： $f[i][j][k]+=f[i-1][j+2][k-2]$

2. 两个都放到之前放过一个棋子的一列中： $f[i][j][k]+=f[i-1][j][k+2]$

3. 一个放到没放过棋子的一列，一个放到放过棋子的一列： $f[i][j][k]+=f[i-1][j+1][k]$

按照乘法原理在转移时乘上该情况的方案数，即可得到答案。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 05
*文件类型: 源代码文件
*题目来源: COGS
*当前状态: 已通过
*备忘录: 动态规划
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 101
#define MOD 9999973
int n,m,f[MAXN][MAXN][MAXN];
/* Variable explain:

*/
int mul(int a,int b)    {return (long long)a*b%MOD;}
int add(int a,int b)    {return (a+b)%MOD;}
int pow(int x,int n)
```

```

{
    int ans=1;
    while(n)
    {
        if(n&1)ans=mul(ans,x);
        x=mul(x,x);
        n>>=1;
    }
    return ans;
}
int C(int a,int b)
{
    if(b==1)return a;
    if(b==2)return a*(a-1)/2;
    int ans=1;
    for(int i=a-b+1;i<=a;++i)
        ans=mul(ans,i);
    for(int i=1;i<=b;++i)
        ans=mul(ans,pow(i,MOD-2));
    return ans;
}
void read()
{
    // freopen("cchess.in","r",stdin);
    // freopen("cchess.out","w",stdout);
    // freopen("in","r",stdin);
    // freopen("out","w",stdout);
    scanf("%d%d",&n,&m);
    if(n<m)swap(n,m);
    return;
}
int main()
{
    read();
    for(int i=m-2;i<=m;++i)
        f[1][i][m-i]=C(m,min(i,m-i));
    for(int i=2;i<=n;++i)
    {
        for(int j=0;j<=m;++j)
        {
            for(int k=0;k<=m-j;++k)
            {
                //j=0,k=1
                f[i][j][k]=f[i-1][j][k];
                if(k) f[i][j][k]=add(f[i][j][k],mul(f[i-1][j+1][k-1],C(j+1,1)));//
0补为1

                if(m-j-k>0) f[i][j][k]=add(f[i][j][k],mul(f[i-1][j]
[k+1],C(k+1,1)));//1补为2

                if(k>1) f[i][j][k]=add(f[i][j][k],mul(f[i-1][j+2][k-2],C(j+2,2)));//
两个0补为1

                if(m-j-k>1) f[i][j][k]=add(f[i][j][k],mul(f[i-1][j]
[k+2],C(k+2,2)));//两个1补为2

                if(m-j-k>0&&k) f[i][j][k]=add(f[i][j][k],mul(f[i-1][j+1]
[k],mul(C(j+1,1),C(k,1))));//各补一个
            }
        }
    }
    int ans=0;
    // for(int u=1;u<=n;++u)
    // {

```

```
// for(int i=0;i<=m;++i)
//     for(int j=0;j<=m-i;++j)printf("%d %d %d %d\n",i,j,m-i-j,f[u][i][j]);
// puts("");
// }
for(int i=0;i<=m;++i)
{
    for(int j=0;j<=m-i;++j)ans=add(ans,f[n][i][j]);
}
printf("%d\n",ans);
return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)