POJ 2411

在$h\times w$的矩阵中铺1×2或2×1的方块，要铺满，问方案数。

题解：看到状压tag了。我的状压是用1表示这一块是2×1的上面那块，需要向下延伸，0表示其他。

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<cstdlib>
using namespace std;
int h,w;
long long f[12][(1<<12)];
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0'||c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
int main()
{
    h=read(),w=read();
    while(h&&w)
    {
        if((h*w)%2){puts("0"),h=read(),w=read();continue;}
        memset(f,0,sizeof(f));
        if(!h&&!w)break;
        for(int i=1;i<=h;i++)
        {
            for(int j=0;j<(1<<w);j++)
            {
                if(i==1)
                {
                    f[i][j]=1;
                    for(int l=1;l<(1<<w);l<<=1)
                    {
                        if((j&l)==0)
                        {
                            if((l<<1)>=(1<<w)){f[i][j]=0;break;}
                            else if(j&(l<<1)){f[i][j]=0;break;}
                            else l<<=1;
                        }
                    }
                }
                else
                {
                    for(int k=0;k<(1<<w);k++)
                    {
                        if(!f[i-1][k])continue;
```

```
                            f[i][j]+=f[i-1][k];
                            for(int l=1;l<(1<<w);l<<=1)
                            {
                                if(k&l)
                                {
                                    if(j&l){f[i][j]-=f[i-1][k];break;}
                                }
                                else if((j&l)==0)
                                {
                                    if((l<<1)>=(1<<w)){f[i][j]-
=f[i-1][k];break;}
                                    else if(j&(l<<1)){f[i][j]-=f[i-1][k];break;}
                                    else if(k&(l<<1)){f[i][j]-=f[i-1][k];break;}
                                    else l<<=1;
                                }
                            }
                        }
                    }
                }
            }
        printf("%lld\n",f[h][0]);
        h=read(),w=read();
    }
    return 0;
}
```

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:weekly:poj_2411&rev=1588997307

Last update: **2020/05/09 12:08**