## 34. Little Bishops

34题这个题很有意思啊，我看了好久终于看懂啦。现在分享一下。

设m\*m的棋盘，放n个象。

首先我们需要把棋盘旋转45°，这样能方便本题的理解。

我们先把棋盘分为两部分，如图，一部分为白色的格子，另一部分是黑色的格子。显而易见，原本在白色格子上面放的象是绝对无法走到黑色的格子上的，黑色也同理无法影响到白色。所以做这道题的第一步，就是把棋盘分为两个部分，这两个部分无法相互影响。最后的答案，就是把这两个部分“拼”起来。

设最终答案为ans，把k个棋子放在白色部分的方案数是w[k](0<=k<=n)，把k个棋子放在黑色部分的方案数是b[k](0<=k<=n)，则ans=∑(b[i]\*w[n-i])，(0<=i<=n)。

所以下一步，就是求每种情况下把k个棋子放在棋盘上的方案数。接下来就需要dp了。

设i是所用行数(就是将地图旋转后的斜线行，每次只考虑同一种颜色的格子)，j是所放的棋子个数，

dp[i][j]就是用i行放j个棋子个数的方案数。同一行显然只能最多放一个棋子。

则dp[i][j]=dp[i-1][j]+dp[i-1][j-1]\*(a[i]-(j-1));其中a[i]是第i行所拥有的格子数。

这个式子其实很好理解。对于dp[i-1][j]，它就是在i-1行放j个棋子的方案数，这一种放法就是在第i行不放棋子；而后一项，就是在第i行放1个棋子的方案数。它的方案数就等于在i-1行放j-1个棋子的方案数，

即dp[i-1][j-1],再乘以在这一行放一个棋子的方案数。这个数就是a[i]-(j-1)，a[i]是格子数，而j-1是之前j-1行所放的棋子数（之前放的每一个棋子都会使现在的方案个数减1）。

这样的话，我们就可以解决绝大多数情况了。但是还有一个大问题。就是棋盘上的格子它的分步数是先递增再递减的。比如说一个8\*8的棋盘，对于白色的格子，它每行的个数分别为1,3,5,7,7,5,3,1，先增后减，对于黑色格子也同理。所以我们会发现，在格子数开始递减时，我们的dp方程会崩溃。它的a[i]数将小于j-1数，所以减完会得到一个负数。所以这时候应该怎么办呢？

我们发现一旦a[i]数开始递减，dp方程就会失效。所以，我们完全可以将每行的格子数排个序(这一步极其机智)。假设8\*8的棋盘，我们可以先dp白格子的第1行，然后第8行；然后第二行，第七行；然后3,6，然后4,5。可以证明，只要当格子数不会递减，dp方程就是成立的，所以我们完全可以不按1,3,5,7,7,5,3,1的顺序来，而是按照1,1,3,3,5,5,7,7的顺序。这样最后算出来的方案数就会是正确的。对于黑色格子，计算方法是一样的。

还有一个初始化的小问题，对于dp[i][0]，应该为1，将不放棋子也视为一种方案；还有，在计算黑色格子的方案数时，dp[1][1]应该为2，因为黑色格子最开始的个数就是从2开始的。

在这里给一个例子，比如说3\*3的棋盘：

对于白色的格子，每行格子数为1,3,1；对于黑色的为2,2；

dpw[1][0]=1;dpw[1][1]=1;

dpw[2][0]=1;dpw[2][1]=dp[1][1]+dp[1][0]\*(1-(1-1))=2;

//这里就是用第一行和第三行排列出来的情况，也就是两个单个格子；

dpw[2][2]=0//用同一条线上面的格子是放不了两个棋子的，故方案数为0；

dpw[3][0]=1;dpw[3][1]=dp[2][1]+dp[2][0]\*(3-(1-1))=5;//一共5个格子，放5个棋子共5种方案

dpw[3][2]=dp[2][2]+dp[2][1]\*(3-(2-1))=4，//2\*2=4

dp[3][3]=dp[2][3]+dp[2][2]\*(3-(2-1))=0。

最后的dp结果：

      0    1    2    3

1    1    1    0    0

2    1    2    0    0

3    1    5    4    0

对于黑色格子：方法同上，只是要将dp[1][1]初始化为2；

       0    1    2    3

1    1     2    0    0

2    1     4    2    0

3    0     0    0    0

当放1个棋子时，答案是5\*1+1\*4=9；

当放2个棋子时，答案是1\*2+5\*4+4\*1=26；

当放3个棋子时，答案是1\*0+5\*2+4\*4+0\*1=26；

当放4个棋子时，答案是2\*4=8（两部分分别放2个才有结果，其他都为0）。