[**状态DP~**](http://www.cppblog.com/notonlysuccess/archive/2009/07/12/89874.html)

<http://acm.sgu.ru/problem.php?contest=0&problem=222>  
这是入门题，数据较大，需要记忆化搜索  
  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1321>  
上题的提高版，不过数据超小，爆搜都能过  
  
<http://acm.sgu.ru/problem.php?contest=0&problem=223>  
先要预处理出一行中的全部可行状态~  
然后DP的时候巧妙的运用位运算进行状态的判断和转移  
状态dp中位运算的巧妙运用会大幅度提高程序的效率和帅气程度  
  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1185>  
非常经典的状态DP，由于攻击范围是两格，所以要保持两个状态，有人用三进制压缩，我觉得太烦了(不能使用飘逸的位运算）  
但是[101][2^10][2^10]得状态太大，考虑到2^10中有很多情况是不可到达的  
计算下当m=10的时候最多60个合法状态，所以我开了[101][60][60]的数组记忆化DP过了  
  
<http://acm.hdu.edu.cn/showproblem.php?pid=2640>  
teddy大牛的题目，和上题差不多，不过不能重叠放，所以处理比上题烦很多  
同样2^8里有很多不可到达的情况，最多之有13种  
所以我开[101][13][13]的数组15ms就过了，哈哈  
这就好像是两次状态压缩  
最近的DP题目感觉到把很多不可到达的状态压缩掉效率会提高超多~也可能让程序从TLE MLE变成AC~  
  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=2411>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1400>  
这道其实很简单，先预处理出当前状态s1到下一状态的可能值s2，hash[1<<m,1<<m]记录，m为较小值  
dp[0][(1<<m)-1] = 1  
然后经过n\*(1<<m)\*(1<<m)的循环得出结果dp[n][(1<<m)-1]  
  
<http://acm.sgu.ru/problem.php?contest=0&problem=223>  
两种砖块，除了预处理的时候状态多点，有7种分支，其他的都和上一题一样  
(主意一个状态到另一个状态可能会有多种情况，hash的时候要用++而不是true false)  
  
<http://acm.hdu.edu.cn/showproblem.php?pid=2280>  
要求用最少的1铺满所有的空格，其中3是没用的(可以用两个5代替)，化简之后使用的方块和上一题一样，一样的预处理后  
dp求出最少的1  
  
<http://acm.pku.edu.cn/JudgeOnline/problem?id=1038>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2696>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2442>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1755>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1820>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1668>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2518>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1666>  
<http://acm.hdu.edu.cn/showproblem.php?pid=1820>  
<http://acm.hdu.edu.cn/showproblem.php?pid=2315>