# 插板法：

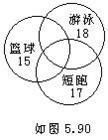
模型1:机器人走n\*m的方格，由左上到右下只能向右或者向下:

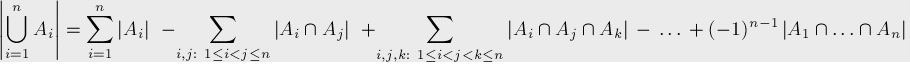
走的每步，要么横着要么竖直走，但一定n个竖着m个横着。江不是分成两种看，用组合数找到n或者m的每种取法。

模型2:任取从n个非负整数，使得和是m的方案数

# 容斥原理及较复杂的问题

<https://blog.csdn.net/werkeytom_ftd/article/details/74701513>

如图所示求三个圆覆盖区域面积，做法是求三个圆面积在减掉中间重复计算的，这种方法就是容斥原理，很容易理解。但是这方面有一些较复杂的问题。所以例题中，尽量选择一些不相同而且有一定复杂度的问题。



容斥原理可以描述如下：

   要计算几个集合并集的大小，我们要先将所有**单个集合**的大小计算出来，然后减去所有**两个集合相交**的部分，再加回所有**三个集合相交**的部分，再减去所有**四个集合相交**的部分，依此类推，一直计算到**所有集合相交**的部分。

容斥原理 相当于先找到所有部分和，不管是否相交。因为算多了要减掉**两个集合相交。**但两两相交中可能再相交所以减多了，又要加回来。之后又加多，再减掉，直到没有nn相交的集合。

组合数容斥模型：

f表示容斥系数。 在一般的问题中，f都表示-1的多少次方。   
现在如果把满足至少1个条件改成至少k个条件，可能会有很多人不知道怎么做了。 但如果你明白容斥的原理，你发现就是 =[n>=k]

知道这个又怎么样呢？   
如果允许n^2的复杂度，你根本不需要去想f是多少，你可以让程序帮你算！   
你需要构造这个f来满足式子。   
我们考虑让n<=m都满足，那么会确定f的前m项。   
现在考虑让n<=m+1都满足，就会将f的第m+1项确定。   
这个f一定是存在的，而且是可以推的。 写一个段代码预处理这个f当然不困难。 同样我们可以思考更加丧病的容斥问题：   
现在对于每个排列，如果有k个位置满足错排，则价值定义为a[k]，求所有排列的价值和。   
假如你明白了容斥的原理，你仍然可以用容斥的思路去解决这题，你只需构造f来满足：   
=a[n]   
这样就很方便了！   
于是我们现在给一道例题：   
这是“玲珑杯”线上赛 Round #17 河南专场某题 。   
有n个硬币，初始全部正面朝上。 现在有m次操作，每次把编号是x的倍数的硬币翻面。 最后问多少个硬币正面朝上。   
那这个题显然也是容斥题，你先用指数级时间枚举操作，然后求个最小公倍数，接下来问题在于容斥系数怎么定。   
有了上面的经验就是构造f满足   
=ord(n)这个f用n^2求没有问题  
例1：hdu6314

N\*M矩阵染色黑白色，至少有a行和b列黑色。问有多少染法

考虑到至少有a行和b列黑色且数据N M a b是3000，对于 j∈[b,M],i∈[a,N] ，C(N,i)代表从N行里选i行黑色的数量 , C(M,j)从M列选j列黑色的数量

那么没被都图成黑色的行列就是, N- i和M- j个

不考虑重复的总数量是

ans=\*2^((N- i)\*( M- j))

上面式子很好理解表示什么都不想，无脑的忽略”重复情况”的情况下，固定某几行和某几列一定是黑色的，其余格子随便染染成什么样都可以，竖着有i∈[a,N], 横着有j∈[b,M] 的染色方法下(每种I,j下,其余格子随便染，由于题意说只有黑白2中颜色，其余格子的方案数是2^((N- i)\*( M- j)) )，染色种类的总和就是枚举i和j，来加和。

但是上面的方法会存在问题是，多算了很多情况(不列举了,自己想)。考虑容斥原理，想办法表示出ans=\*2^((N- i)\*( M- j))的多于部分

既然是多于部分，对于每一块i和j一定是在基础搞，对于每个i和j可以枚举剩余行和剩余列，即：ii∈[0,N-i] ,jj∈[0,M-j] 那么就是

令n=N-i,m=M-j

对于ii=0且jj=0的情况下。表示了从n和m的区域里随便涂色。我们要排除有些行或者列都是黑色的情况。利用组合数容斥公式

f(i)是容斥系数，它是由另一维m决定，另一维是

g(j)是第二个容斥系数，它是由这2维共同决定,(-1)^(i+j) ,我也不是太懂

对了这里还不要忘了乘2^((n-i)\*(m-j)) 因为有这么多种添法，在其余的格子里

所以总公式是： 再套一个：

\*2^((n- ii)(m- jj))\* 2^((n-ii)\*(m-jj))

到这里还没结束，这个公式复杂度太高，动手整理成n^2的是：