四．多项式定理：

I.i.在的展开式中，对任意给定的项【其中】，它的系数为，与n无关【比如，中的项系数与中的项系数相同】。

【它的得来方法：m个不同的格子里已经放好了m个给定的ai(可能有相同的ai)，现在让这些ai们在这m个格子之间疯狂地轮换位置，有多少种不重复的组合(而不是排列)就有多少个对应的不重复的[给定项]的得来途径】

ii. 在的展开式中，对任意给定的项【其中】，它的所有同类项的个数为【“同类项”是指，指数bi们在组合(不要求一定要在排列上)上均为相同的某特定数组[b1,b2,b3,···,bn]的项们；比如与】。

【它的得来方法：n个不同的ai放入n个以bi为标签的格子里(可能有相同的bi)，现在让这些ai们在这n个格子之间疯狂地轮换位置，有多少不重复的组合(而不是排列)就有多少个同类项】【即使可能有些ai 的值相同，这里仍然以i为标签,视为不同的ai】

II.i.，其中。【我发现不因某一个求和符号而有关系的两个i便不是同一个i，因而可以有两个相同长相的i身处两个不关联的求和计算中，这样省略了寻找新的自变量外形的困扰。】其中[?]

=[把m写作n个数字xi(1≤i≤n，0≤xi≤m)的和时，所有不重复的(有效的)组合的个数]。=像这样把m个球投入以n为长度的底的45°斜二维方桶里，球球们共有多少种可能的分布方式 。【重力加速度竖直向下】

=像这样把m个球投入以n为长度的底的45°斜二维方桶里，球球们共有多少种可能的分布方式 。【重力加速度竖直向下】

=[把m写作m个数字xi(1≤i≤m，0≤xi≤n)的和时，所有不重复的(有效的)组合的个数]

【注：“?”≠单纯的隔板法得出的结果；这里我简单介绍一下啥子叫隔板法以及它的适用范围/对象：对=[把M写作n个数字xi(1≤i≤n，1≤xi≤M-(n-1))的和时，所有不重复的(有效的)排列的个数]=[把M-n写作n个数字xi(1≤i≤n，0≤xi≤M-(n-1)-1)的和时，所有不重复的(有效的)排列的个数]，

现在令M-(n-1)-1=m，则有M=m+n，则=[把m写作n个数字xi(1≤i≤n，0≤xi≤m)的和时，所有不重复的(有效的)排列的个数]，此时就和[?]=[把m写作n个数字xi(1≤i≤n，0≤xi≤m)的和时，所有不重复的(有效的)组合的个数]很接近了。但排列与组合一词之差，让他俩关系又相去甚远。[不过的展开式与的展开式可以说是高度相似的：最外层都是对某种分类方式下的特征值从z=1到?的求和，其中?的属性为组合]】

ii.受斜方桶的启发，在实在找不出那个纯粹精炼的数学表达式以及得到它的任何途径的情况下，我开发了一种算法：m~n算法，让计算机可以逻辑鲜明地累加得到最终结果，这种算法其实就是模拟穷举过程，但其价值就确实在于“有序地穷举”：

1. 由于即使n>m，n也相当于m，所以现记m~n为m~min(m,n)。现把m个球放在以n为底的限制(宽的)桶里，并占满最下一层(记为第1层)，min(m,n)个球；此时第1层以上所有层里的m-min(m,n)个球的总组合数记为=m-min(m,n)~ min(m,n)=m-min(m,n)~min(m-min(m,n),min(m,n))。

2.把第1层的球数min(n,m)减去1，并使上面所有层的球数不超过min(n,m)-1(即相当于设定了新的n值=n-1)，此时1层以上所有层的球的总组合数记为=m- min(m,n)+1~min(m,n)-1=m-min(m,n)+1~min(m-min(m,n)+1,min(m,n)-1)。

3.知=m-min(m,n)+i~min(m,n)-i=m-min(m,n)+i~min(m-min(m,n)+i, min(m,n)-i)，则进一步可知m~n=m~min(m,n)==

4.把每个i所对应的m-min(m,n)+i~min(m-min(m,n)+i, min(m,n)-i)都看作下一个新的m~min(m,n)，进行分解直到不能分解为止。其中不能分解的标志是：若某一级m~min(m,n)尝试着继续分解，当i=1[0]时，m-min(m,n)+i~min(m-min(m,n)+i, min(m,n)-i)中的min(m-min(m,n)+i, min(m,n)-i)中的min(m,n)-i≤0[1]，即m~min(m,n)中的min(m,n) ≤1，此时就返回上一级，即表明该级已不能再被分解。

5.计数所有末端/末梢/最内层/最外围/最底端/最低端的m~1和0~0，其个数就等于我们要求的[?]值。那么下面我们来举例模拟一下下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7~5=下列5个彩色的和(1) | 2~2=下列2个彩色的和(1.1) | 3~3=下列3个彩色的和(1.2) | 4~3=下列3个彩色的和(1.3) | 2~2=下列2个彩色的和(1.3.1) | 5~2=下列2个彩色的和(1.4) | 3~2=下列2个彩色的和(1.4.1) |
| 2~5=2~2 | 0~2=0~0 |  |  |  |  |  |
| 3~4=3~3 | 1~1=1~1 | 0~3=0~0 |  |  |  |  |
| 4~3=4~3 |  | 1~2=1~1 | 1~3=1~1 |  |  |  |
| 5~2=5~2 |  | 2~1=2~1 | 2~2=2~2 | 0~2=0~0 | 3~2=3~2 | 1~2=1~1 |
| 6~1=6~1 |  |  | 3~1=3~1 | 1~1=1~1 | 4~1=4~1 | 2~1=2~1 |

【注释：该表格中所有红色代表不可分解的分解值，即最后的有效计数单位，共计13个，则[?]=13；所有黑色代表运算后但没化简的分解值；所有蓝色代表化简后的可分解的分解值，蓝色由深到浅7~5→7~5→7~5代表化简后的可进一步分解的分解值，颜色越浅表示层数更内，同颜色表示同层数；数字1.3.1表示：7~5这个第一个可分解的值分解后，第二层可分解值中的第三个可分解值4~3分解后，第三层可分解值中该值4~3所代表的分支下的第一个可分解值2~2】

【现在各位便知道为什么我要费力地每一步都引入n=min(m,n)这个赋值操作了吧，因为只有这样真实的n值，才是判断是否可以被继续分解的标志】

【接着我们又可从下式m~n=m~min(m,n)=看出每一个新的m~min(m,n)的分解产物总数为min(m,n)-1+1个，即min(m,n)个，那么[?]= = ==包括初始对象7~5在内的所有有效n值min(m,n)不是0或1的可分解值的有效n值min(m,n)之和不包括初始对象7~5在内的所有所有有效n值min(m,n)不是0或1的可分解值的个数之和=包括初始对象7~5在内的所有有效n值min(m,n)不是0或1的可分解值的有效n值min(m,n)之和其个数之和+1 例：对于7~5来说，其[?]=(5+2+3+3+2+2+2)-7+1=13。对计算机来讲如果可以更快的话[?]=(5-1+2-1+3-1+3-1+2-1+2-1+2-1)+1=13也可】