# 母函数（Generating function）详解

在数学中，某个序列的****母函数(Generating function，又称**[生成函数](https://www.baidu.com/s?wd=%E7%94%9F%E6%88%90%E5%87%BD%E6%95%B0&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/qq_41603898/article/details/_blank)**)****是一种形式幂级数，其每一项的系数可以提供关于这个序列的信息。使用母函数解决问题的方法称为****母函数方法****。

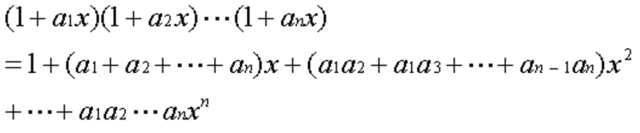
母函数可分为很多种，包括**[普通母函数](http://www.wutianqi.com/?p=596" \t "https://blog.csdn.net/qq_41603898/article/details/_blank)**、**[指数母函数](http://www.wutianqi.com/?p=2644" \t "https://blog.csdn.net/qq_41603898/article/details/_blank)**、****L级数****、****贝尔级数****和****狄利克雷级数****。对每个序列都可以写出以上每个类型的一个母函数。构造母函数的目的一般是为了解决某个特定的问题，因此选用何种母函数[视乎](https://www.baidu.com/s?wd=%E8%A7%86%E4%B9%8E&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/qq_41603898/article/details/_blank)序列本身的特性和问题的类型。

这里先给出两句话，不懂的可以等看完这篇文章再回过头来看：

1.“把组合问题的加法法则和幂级数的乘幂对应起来”

2.“母函数的思想很简单 — 就是把离散数列和幂级数一 一对应起来，把离散数列间的相互结合关系对应成为幂级数间的运算关系，最后由幂级数形式来确定离散数列的构造. “

我们首先来看下这个多项式乘法：



母函数图(1)

由此可以看出:

1.x的系数是a1,a2,…an 的单个组合的全体。

2. x^2的系数是a1,a2,…a2的两个组合的全体。

………

n. x^n的系数是a1,a2,….an的n个组合的全体（只有1个）。

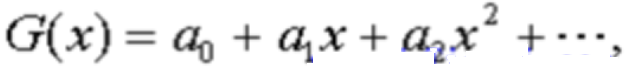
进一步得到：



母函数图(2)

****母函数的定义****

对于序列a0，a1，a2，…构造一函数：



母函数图(3)

称函数G(x)是序列a0，a1，a2，…的母函数。

这里先给出2个例子，等会再结合题目分析：

****第一种：****

有1克、2克、3克、4克的砝码各****一枚****，能称出哪几种重量？每种重量各有几种可能方案？

考虑用母函数来解决这个问题：

我们假设x表示砝码，x的****指数****表示砝码的重量，这样：

1个1克的砝码可以用函数1+1\*x^1表示，

1个2克的砝码可以用函数1+1\*x^2表示，

1个3克的砝码可以用函数1+1\*x^3表示，

1个4克的砝码可以用函数1+1\*x^4表示，

上面这四个式子懂吗？

我们拿1+x^2来说，前面已经说过，x表示砝码，x的指数表示砝码的重量！初始状态时，这里就是一个质量为2的砝码。

那么前面的****1****表示什么？按照上面的理解，1其实应该写为：1\*x^0,即1代表重量为2的砝码数量为0个。

所以这里1+1\*x^2 = 1\*x^0 + 1\*x^2，即表示2克的砝码有两种状态，不取或取，不取则为1\*x^0，取则为1\*x^2

不知道大家理解没，我们这里结合前面那句话：

“****把组合问题的加法法则和幂级数的乘幂对应起来****“

接着讨论上面的1+x^2，这里x前面的系数有什么意义？

这里的系数表示****状态数****(****方案数****)

1+x^2，也就是1\*x^0 + 1\*x^2，也就是上面说的不取2克砝码，此时有1种状态；或者取2克砝码，此时也有1种状态。(分析！)

所以，前面说的那句话的意义大家可以理解了吧？

几种砝码的组合可以称重的情况，可以用以上几个函数的乘积表示：

(1+x)(1+x^2)(1+x^3)(1+x^4)

=(1+x+x^2+x^4)(1+x^3+^4+x^7)

=1 + x + x^2 + 2\*x^3 + 2\*x^4 + 2\*x^5 + 2\*x^6 + 2\*x^7 + x^8 + x^9 + x^10

从上面的函数知道：****可称出从1克到10克，系数便是方案数。（！！！经典！！！）****

例如右端有2^x^5 项，即称出5克的方案有2种：5=3+2=4+1；同样，6=1+2+3=4+2；10=1+2+3+4。

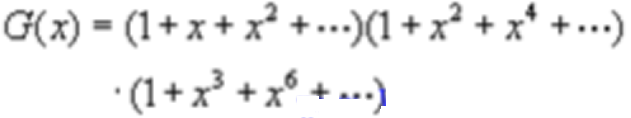
故称出6克的方案数有2种，称出10克的方案数有1种 。

接着上面，接下来是第二种情况：

****第二种：****

求用1分、2分、3分的邮票贴出不同数值的方案数：

大家把这种情况和第一种比较有何区别？第一种每种是****一个****，而这里每种是****无限****的。



母函数图(4)

以展开后的x^4为例，其系数为4，即4拆分成1、2、3之和的拆分方案数为4；

即 ：4=1+1+1+1=1+1+2=1+3=2+2

这里再引出两个概念"整数拆分"和"拆分数"：

所谓****整数拆分****即把整数分解成若干整数的和（相当于把n个无区别的球放到n个无标志的盒子，盒子允许空，也允许放多于一个球）。

整数拆分成若干整数的和，办法不一，不同拆分法的总数叫做****拆分数****。

现在以上面的第二种情况每种种类个数无限为例，给出****模板****：

#include <bits/stdc++.h>

*//1~n\_Num 无限组成n\_Num的方法数*

using namespace std;

const int maxn=10001;

int c1[maxn],c2[maxn];

*//c1保存各种组合的方法数*

*//c2是中间变量*

int main()

{

int n\_Num,i,j,k;

while(~scanf("%d",&n\_Num))

{

for(int i=0;i<=n\_Num;i++)*//初始化第一个式子*

{*//（1+x+x^2+x^3+...+x^n）各种质量方法数都是1*

c1[i]=1;

c2[i]=0;

}

for(int i=2;i<=n\_Num;i++)*//从第二个式子开始*

{

for(int j=0;j<=n\_Num;j++)*//j表示前面累乘的式子的第j个数*

*//对于(1+x)(1+x^2)(1+x^3)*

*//i=2 指行完成后 （1+x+x^2+x^3）j就指向这个式子*

*//c1指向累乘的式子的系数 c2是新的式子*

for(int k=0;k+j<=n\_Num;k+=i)*//第i个式子每次增量是i,所以加i*

{*//这个k指第j个数的指数*

c2[k+j]+=c1[j];

}

for(int j=0;j<=n\_Num;j++)

{

c1[j]=c2[j];

c2[j]=0;

}

}

printf("%d\n",c1[n\_Num]);

}

return 0;

}

                         (\*\*\*\*\*\*\*\*\*\*\*****！！！重点！！！****\*\*\*\*\*\*\*\*\*\*\*)

 5个for循环的意义

①  、首先对c1初始化，由第一个表达式(1+x+x^2+..x^n)初始化，把质量从0到n的所有砝码都初始化为1.

②  、 i从2到n遍历，这里i就是指第i个表达式，上面给出的第二种母函数关系式里，每一个括号括起来的就是一个表达式。

③、j 从0到n遍历，这里j就是(前面i個表达式累乘的表达式)里第j个变量，(这里感谢一下seagg朋友给我指出的错误，大家可以看下留言处的讨论)。如(1+x)(1+x^2)(1+x^3)，j先指示的是1和x的系数，i=2执行完之后变为  
  
（1+x+x^2+x^3）(1+x^3)，这时候j应该指示的是合并后的第一个括号的四个变量的系数。

④ 、 k表示的是第j个指数，所以k每次增i（因为第i个表达式的增量是i）。

⑤  、把c2的值赋给c1,而把c2初始化为0，因为c2每次是从一个表达式中开始的。