ACM 程序设计

计算机学院 刘春英

今天，

你 AC 了吗？

每周一星（9）：

ForYou

西安赛区汇报…

第十讲

母函数及其应用
（Generation function）

从递推关系说起

研究以下等式：

$$
\begin{aligned}
&(1+a_1x)(1+a_2x)\cdots(1+a_nx)\\
&=1+(a_1+a_2+\cdots+a_n)x+(a_1a_2+a_1a_3+\cdots+a_{n-1}a_n)x^2\\
&+\cdots+a_1a_2\cdots a_nx^n \qquad (2\text{-}1\text{-}1)
\end{aligned}
$$

可以看出：

x2项的系数a1a2+a1a3+...+an-1an中所有的项包括n个元素a1，a2，…an中取两个组合的全体；同理x3项系数包含了从n个元素a1，a2，…an中取3个元素组合的全体。以此类推。

若令a1=a2= …=an=1，在（2-1-1）式中a1a2+a1a3+…+an-1an项系数中每一个组合有1个贡献，其他各项以此类推。故有：

$$(1+x)^n = 1 + C(n,1)x + C(n,2)x^2 + \cdots + C(n,n)x^n \qquad (2\text{-}1\text{-}2)$$

母函数定义：

- 对于序列a0，a1，a2，…构造一函数：

$$G(x) = a_0 + a_1 x + a_2 x^2 + \cdots,$$

- 称函数G(x)是序列a0，a1，a2，…的母函数

For example:

(1+x)n是序列C(n,0),C(n,1),...,C(n,n)的母函数。

如若已知序列a0，a1，a2，…则对应的母函数G(x)便可根据定义给出。

反之，如若已经求得序列的母函数G(x)，则该序列也随之确定。

序列a0，a1，a2，…可记为{an}。

实例分析

例一、若有1克、2克、3克、4克的砝码各一枚，能称出哪几种重量？各有几种可能方案？

如何解决这个问题呢？考虑构造母函数。

如果用x的指数表示称出的重量，则：

1个1克的砝码可以用函数1+x表示，

1个2克的砝码可以用函数1+x2表示，

1个3克的砝码可以用函数1+x3表示，

1个4克的砝码可以用函数1+x4表示，

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：

- $(1+x)(1+x2)(1+x3)(1+x4)$
- $=(1+x+x2+x3)(1+x3+x4+x7)$
- $=1+x+x2+2x3+2x4+2x5+2x6+2x7+x8+x9+x10$
-
- 从上面的函数知道，可称出从1克到10克，系数便是方案数。例如右端有2x5 项，即称出5克的方案有2：5=3+2=4+1，同样，6=1+2+3=4+2；10=1+2+3+4。故称出6克的方案有2，称出10克的方案有1

例二、求用1分、2分、3分的邮票贴出不同数值的方案数。

- 因邮票允许重复，故母函数为

$$G(x)=(1+x+x^2+\cdots)(1+x^2+x^4+\cdots)\cdot(1+x^3+x^6+\cdots)$$

- 以展开后的x4为例，其系数为4，即4拆分成1、2、3之和的拆分数为4，即：
- 4=1+1+1+1=1+1+2=1+3=2+2

概念：整数拆分

- 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，盒子允许空着，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。

练习：

- 例3：若有1克砝码3枚、2克砝码4枚、4克砝码2枚，问能称出哪几种重量？各有几种方案？
- 例4: 整数n拆分成1，2，3，…，m的和，求其母函数。如若其中m至少出现一次，其母函数又如何？

- 请自己写出以上两个问题的母函数。

如何编写程序
实现母函数的应用呢？

关键：对多项式展开

以整数拆分为例：

观察以下的母函数：

$$G(x) = (1 + x + x^2 + \cdots)(1 + x^2 + x^4 + \cdots) \cdot (1 + x^3 + x^6 + \cdots)$$

（这里以lwg写的程序为例）

```
#include <iostream>
using namespace std;
const int lmax=10000;
int c1[lmax+1],c2[lmax+1];
int main()
{     int n,i,j,k;
      while (cin>>n)
      {for (i=0;i<=n;i++)
       {c1[i]=0;     c2[i]=0;       }
       for (i=0;i<=n;i++) c1[i]=1;
       for (i=2;i<=n;i++)
          {        for (j=0;j<=n;j++)
                        for (k=0;k+j<=n;k+=i)
                         {  c2[j+k]+=c1[j];  }
                   for (j=0;j<=n;j++)
                   {  c1[j]=c2[j];        c2[j]=0;  }
          }
          cout<<c1[n]<<endl;
      }
      return 0;
}
```

HDOJ 1398 Square Coins

Sample Input

2
10
30
0

Sample Output

1
4
27

算法分析：

典型的利用母函数可解的题目。

G(x
)=(1+x+x2+x3+x4+…)(1+x4+x8+x12+…)(1+x9+x18+x27+…)…

```
#include <iostream>
using namespace std;
const int lmax=300;
int c1[lmax+1],c2[lmax+1];
int main(void)
{   int n,i,j,k;
    while (cin>>n && n!=0)
    {       for (i=0;i<=n;i++)
            {       c1[i]=1;        c2[i]=0;            }
            for (i=2;i<=17;i++)
            {       for (j=0;j<=n;j++)
                            for (k=0;k+j<=n;k+=i*i)
                            {       c2[j+k]+=c1[j];     }
                    for (j=0;j<=n;j++)
                    {       c1[j]=c2[j];    c2[j]=0;    }
            }
            cout<<c1[n]<<endl;
    }
    return 0;
}
```

```
…
int main(void)
{   int n,i,j,k;
    int elem[17]={1,4,9,16,25,36,…,169,196,225,256,279}
    while (cin>>n && n!=0)
    {       for (i=0;i<=n;i++)
            {       c1[i]=1;        c2[i]=0;            }
            for (i=2; i<=17; i++)
            {       for (j=0;j<=n;j++)
                        for (k=0;k+j<=n; k+=elem[i-1] )
                        {       c2[j+k]+=c1[j];         }
                    for (j=0;j<=n;j++)
                    {       c1[j]=c2[j];    c2[j]=0;    }
            }
            cout<<c1[n]<<endl;
    }
    return 0;
}
```

Any Questions?

思考：

HDOJ_1028

Ignatius and the Princess III

思考：

HDOJ_1085
Holding Bin-Laden Captive!

思考：

HDOJ_1171 Big Event in HDU

思考：

HDOJ_1059

[Dividing]()

后面的安排:

- 2007/01/01 元旦放假
- 2007/01/08 上机练习（模拟考试）
- 2007/01/15 期末考试（如有变化，论坛通知）

- 上机地点：现教317和现教321
- 上机时间：17：05~20：30

新年寄语…

附：相关练习题(hdoj)：

- 1028、1059、1085
- 1171、1398、2069

- 其它相关题目（比如求邮票、硬币之类的组合数、整数的不同拆分数等）