树的计数

华中师大一附中 赵爽

【关键字】

树　图　组合数　二项式定理　母函数　微商　不定积分

【预备知识】

组合数学部分

推广的组合数（二项式系数）：

推广的组合数记作$\binom{r}{k}$。定义为$\binom{r}{k}=\frac{r(r-1)\cdots(r-k+1)}{k(k-1)\cdots 2\cdot 1}=\prod_{1\le j\le k}\left(\frac{r+1-j}{j}\right)$，其中$k\in Z^{+}$。在$r\in Z^{+}$时，$\binom{r}{k}$就等于普通组合数$C_r^k$。

推广组合数的一个性质：

由推广组合数的定义，可以直接得出：

对于所有的$k\in Z$，有$\binom{-r}{k}=(-1)^k\binom{r+k-1}{k}$。

牛顿二项式定理：

$$(x+y)^r=\sum_{k\ge 0}\binom{r}{k}x^k y^{r-k}\text{。}$$

二项式定理的应用 1：

由二项式定理，有：$(1+z)^r=\sum_{k\geq 0}\binom{r}{k}z^k$ 。

当 $r<0$ 时，有：$(1-z)^r=\frac{1}{(1-z)^{-r}}=\sum_{k\geq 0}\binom{-r}{k}(-z)^k=\sum_{k\geq 0}\binom{r+k-1}{k}z^k$ 。如果令 $a=-r$，我们有：$\frac{1}{(1-z^r)^a}=\sum_{j\geq 0}\binom{a+j-1}{j}(z^r)^j=\sum_{j\geq 0}\binom{a+j-1}{j}z^{r\cdot j}$，其中 $a>0$。特别的，在 $a=r=1$ 的时候，有 $\frac{1}{1-z}=1+z+z^2+\cdots$。

二项式定理的应用 2:

对二项式定理应用 1 中的结论 $\frac{1}{1-z}=1+z+z^2+\cdots$ 左右同时求不定积分，得到：

$$\int\frac{1}{1-z}dz=\ln\frac{1}{1-z}=\int(1+z+z^2+\cdots)dz=z+\frac{1}{2}z^2+\frac{1}{3}z^3+\cdots=\sum_{k\geq 1}\frac{z^k}{k}\text{。}$$

图论部分

一个不含回路的图称为**无根树**（或**自由树**）。

一个**有根树**（或**有向树**）是这样一个有着一个确定的顶点 R 的有向图：

a) 每个顶点 $V\neq R$ 恰好是以 $e[V]$ 表示的一条弧的初始顶点；
b) R 不是任何弧的初始顶点；
c) R 在上述定义的意义下是一个根（即，对于每个顶点 $V\neq R$，有一条从 V 到 R 的**唯一**有向通路。因此，整个有根树中不存在圈。

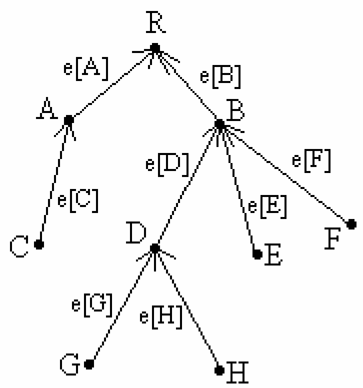
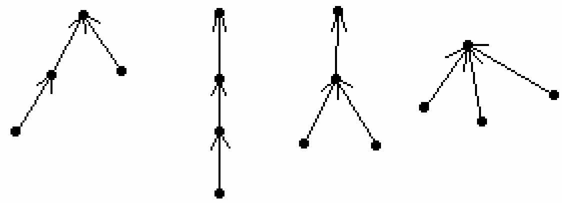


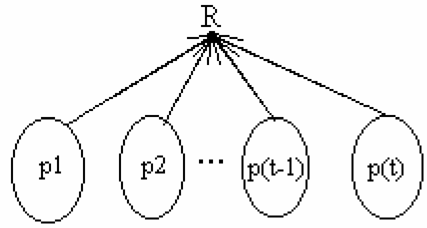
图1 有根树

【问题 1】有根树的计数

用 n 个相同顶点能够构成多少种结构不同的有根树呢？在 $n=4$ 的时候，有下面 4 种构造方法：



我们不妨假设用n个顶点能够构成a_n种有根树。那么，很显然有$a_1=1$。当$n>1$时，这棵树的结构必然是：



即这棵有根树包含t棵子树，第i棵子树包含的顶点个数是p_i。那么，必然有$\sum_{i=1}^{t} p_i = n-1$。

为了便于记数，我们不妨假设含有 1 个顶点的子树有j_1个，含有 2 个顶点的子树有j_2个，……，含有$n-1$个顶点的子树有j_{n-1}个。那么，有$\sum_{i=1}^{j-1} i \cdot j_i = n-1$。

含有k的顶点的子树有j_k个，而这j_k个子树每个均有a_k种构造方法。我们假设这j_k个子树中有x_1个采用的是第 1 种构造方法，有x_2个采用的是第 2 中构造方法，……，有x_{a_k}个采用的是第a_k种构造方法，那么，有$\sum_{i=1}^{a_k} x_i = j_k$。这个方程非负整数解的个数$\binom{a_k+j_k-1}{j_k}$就是这$j_k$个包含$k$个顶点的子树的构造方案总数[①]。

因此，用n个顶点可以构造的有根树总数为：

$$a_n = \sum_{j_1+2j_2+\cdots+(n-1)j_{n-1}=n-1} \left[\prod_{k=1}^{n-1} \binom{a_k+j_k-1}{j_k} \right] \cdots\cdots\cdots\cdots \text{等式 1}$$

当然，这里有$n>1$。

令$a_0=0$，下面我们研究数列$\{a_n\}$的母函数$A(z)=\sum_{n\geq 1} a_n z^n$的性质。由

① 关于方程的非负整数解个数问题，参见有关组合数学书籍。

$\dfrac{1}{\left(1-z^{r}\right)^{a}}=\sum_{j\geq 0}\binom{a+j-1}{j}z^{rj}$（推导已经在“预备知识”部分给出），那么有：

$\prod_{k\geq 1}\left[\dfrac{1}{\left(1-z^{k}\right)^{a_k}}\right]=\prod_{k\geq 1}\sum_{j\geq 0}\left[\binom{a_k+j-1}{j}z^{kj}\right]$。这个计算结果必定是一个 z 的多项式。把这个多项式展开，并分析 z^n 的系数，可以得到 z^n 的系数：$\sum_{j_1+2j_2+3j_3\cdots=n}\left[\prod_{k\geq 1}\binom{a_k+j_k-1}{j_k}\right]$。

再由等式 1，我们发现这里 z^n 的系数恰为 a_{n+1}。因此我们得到了数列 $\{a_n\}$ 的母函数：

$A(z)=\dfrac{z}{(1-z)^{a_1}\left(1-z^2\right)^{a_2}\cdots}$。然而这并不是 $A(z)$ 的的理想表达方式，因为这是一个无限的乘积，而且等式的右边出现了 a_i。因此我们还需要对这种形式作若干变换。

不妨对上面的式子左右取自然对数，有：

$\ln A(z)=\ln z+\sum_{k\geq 1}a_k\ln\left(\dfrac{1}{1-z^k}\right)$。根据“预备知识”部分“二项式定理的应用 2”的结论，我们可以得出：$\sum_{k\geq 1}a_k\ln\left(\dfrac{1}{1-z^k}\right)=\sum_{k\geq 1}\left[a_k\sum_{t\geq 1}\dfrac{\left(z^k\right)^t}{t}\right]=\sum_{k,t\geq 1}\dfrac{a_kz^{kt}}{t}$。因此，有

$\ln A(z)=\ln z+\sum_{k,t\geq 1}\dfrac{a_kz^{kt}}{t}=\ln z+\sum_{t\geq 1}\dfrac{A\left(z^t\right)}{t}$。所以我们得到了 $A(z)$ 的一种比较好的形式：

$$A(z)=z\cdot\exp[A(z)+\frac{1}{2}A\left(z^2\right)+\frac{1}{3}A\left(z^3\right)+\cdots]\ \cdots\cdots\cdots\cdots\ \text{等式 2}$$

将等式 2 左右两边同时对 z 求微商，有：

左边：$\dfrac{\partial A(z)}{\partial z}$

右边：

$$\frac{\partial\left\{z\cdot\exp\left[\sum_{k\geq 1}\frac{A\left(z^k\right)}{k}\right]\right\}}{\partial z}=\exp\left[\sum_{k\geq 1}\frac{A\left(z^k\right)}{k}\right]+z\cdot\exp\left[\sum_{k\geq 1}\frac{A\left(z^k\right)}{k}\right]\cdot\frac{\partial\sum_{i\geq 1}\frac{A\left(z^i\right)}{i}}{\partial z}$$

$$=\frac{A(z)}{z}+A(z)\cdot\sum_{i\geq 1}\left[z^{i-1}\cdot\frac{\partial A\left(z^i\right)}{\partial z^i}\right]$$

因此有：

$$\frac{\partial A(z)}{\partial z}-\frac{A(z)}{z}=A(z)\cdot\sum_{i\geq 1}\left[z^{i-1}\cdot\frac{\partial A(z^i)}{\partial z^i}\right]$$

展开，即得：

$$\sum_{n\geq 1}na_{n+1}z^n=\left(\sum_{k\geq 1}a_kz^k\right)\sum_{i\geq 1}\left\{z^{i-1}\cdot\sum_{j\geq 1}\left[ja_jz^{i\cdot(j-1)}\right]\right\}=\left[\sum_{j\geq 1}\left(ja_j\cdot\sum_{i\geq 1}z^{i\cdot j-1}\right)\right]\left(\sum_{k\geq 1}a_kz^k\right)$$

我们分析一下上面那个等式左右两边 z^n 的系数。等式左边 z^n 的系数为 na_{n+1}，而等式右边的 z^n 是由两个括号各提供一些元素得到。假设第一个括号提供的是 $ja_j\cdot z^{i\cdot j-1}$，那么第二个括号提供的必然是 $a_{n+1-i\cdot j}z^{n+1-i\cdot j}$。因此有：

$$na_{n+1}=\sum_{j\geq 1}\left(ja_j\sum_{1\leq i\leq\frac{n}{j}}a_{n+1-i\cdot j}\right)$$

为了便于记忆和计算，我们对上面的式子稍加变化：

令 $s_{n,j}=\sum_{1\leq i\leq\frac{n}{j}}a_{n+1-i\cdot j}$，则有：$s_{n,j}=s_{n-j,j}+a_{n+1-j}$。因此我们得到了寻求 a_n 比较理想的递推式：

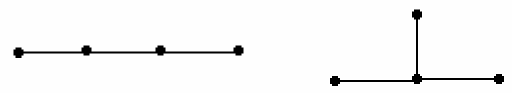
$$a_{n+1}=\frac{\sum_{1\leq j\leq n}ja_js_{n,j}}{n}$$

根据这个递推式，我们就可以求出 $A(z)$：

$$A(z)=z+z^2+2z^3+4z^4+9z^5+20z^6+48z^7+115z^8+286z^9+719z^{10}+1842z^{11}+\cdots$$

【问题 2】无根树的计数

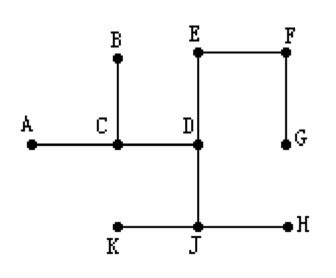
下面我们讨论用 n 各相同顶点能够构成多少无根树呢？比如 $n=4$ 的时候，只有 2 种不同的构造方法：



因为边的方向被去掉之后，问题 1 中的前两种方案和后两种方案各自都只代表 1 种方法了。

我们不难看到，对于一个给定由 n 个顶点构成的无根树，我们可以任意选择一个顶点 X，并以这个顶点为根，用唯一的方式把所有的边赋予方向，使它变成一个以 X 为根的有

根树。一旦这一步已经完成，假设根 X 有 k 个子树，这些子树中分别有 $s_1, s_2, s_3, \cdots, s_k$ 个顶点，而且一定有 $\sum_{i=1}^{k} s_i = n-1$。在这种情况下，我们定义 X 的权为 $\max\{s_1, s_2, \cdots, s_k\}$。于是，在无根树



中，顶点 D 的权是 3，而顶点 E 的权是 7。在一个无根树中，具有极小权值的顶点，称作该树的一个形心。

假设上面定义的 X 和 k 个子树的分别根是 $Y_1, Y_2, Y_3, \cdots, Y_k$，那么以 Y_1 为根的子树中的每一个顶点的权至少是 $n - s_1 = 1 + s_2 + s_3 + \cdots + s_k$。如果在 Y_1 子树中有一个形心 Y，则我们有

$$权(X) = \max(s_1, s_2, \cdots, s_k) \geq 权(Y) \geq 1 + s_2 + \cdots + s_k$$

由 $\max\{s_1, s_2, \cdots, s_k\} \geq 1 + s_2 + s_3 + \cdots + s_k$ 我们可以得出：

$$\max\{s_1 + s_2 + \cdots + s_k\} = s_1 > s_2 + s_3 + \cdots + s_k$$

我们可以把这个结论推广到 Y_j 子树中存在形心的情况。因此对于任何一个顶点 X，它的所有子树中至多有一个包含形心。这是一个很强的条件，因为它意味着一个无根树至多存在两个形心，如果有两个形心，那么它们一定相邻。

反之，如果 $s_1 > s_2 + s_3 + \cdots + s_k$，则在 Y_1 子树中必定存在一个形心。这是因为

$$权(Y_1) \leq \max\{s_1 - 1, 1 + s_2 + s_3 + \cdots + s_k\} \leq s_1 = 权(X)$$

而且在 $Y_2, Y_3, \cdots, Y_k$ 子树中，所有顶点的权至少是 $s_1 + 1$。

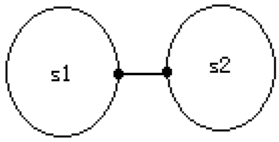
因此我们得到重要结论：

顶点 X 是一个无根树的唯一形心，当且仅当对于所有的 $j\left(1 \leq j \leq k\right)$，有

$$2s_j \leq s_1 + s_2 + \cdots + s_k 。$$

因此具有 n 个顶点，仅有一个形心的无根树的个数，是具有 n 个顶点的有根树个数减去

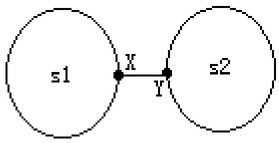
不满足上面条件的有根树的个数。而不满足条件的有根树必然是以下的形式：



其中 $s_2 = n - s_1 \le s_1$。因此不满足条件的有根树的总数必然为：$\sum_{1 \le i \le \frac{n}{2}} a_i a_{n-i}$。我们从而得到了只有一个形心的无根树的个数：

$$a_n - \sum_{1 \le i \le \frac{n}{2}} a_i a_{n-i} \quad \cdots\cdots\cdots\cdots \quad \text{等式 3}$$

下面我们考虑具有两个形心的无根树的个数。由于这两个形心必然相邻（如下图）：



假设 X 和 Y 是这个无根树的两个形心，很明显，$s_1 + s_2 = n$。上面已经说过，如果 Y 是一个形心，则 $\text{权}(X) = s_2$。同样的，有 $\text{权}(Y) = s_1$。因此必然有 $s_1 = s_2 = \frac{n}{2}$。

这样一来，我们就发现，具有两个形心的无根树一定包含偶数个顶点。假设 $m = \frac{n}{2}$，那么构造这样的无根树的方法总数，相当于多重集 $\{\infty \cdot 1, \infty \cdot 2, \infty \cdot 3, \cdots, \infty \cdot a_m\}$ 的 2-组和数，即 $\binom{a_m + 1}{2}$。因此，为了得到无根树的总数，在 n 为偶数时，我们把 $\binom{a_{\frac{n}{2}} + 1}{2}$ 加到等式 3 中，即可。

综上所述，有：

当 n 是奇数时，无根树一共有 $a_n - \sum_{1 \le i \le \frac{n}{2}} a_i a_{n-i}$ 种构造方法。

当 n 是偶数时，无根树一共有 $a_n - \sum_{1 \le i \le n} a_i a_{n-i} + \frac{1}{2} a_{\frac{n}{2}} \left(a_{\frac{n}{2}} + 1 \right)$ 种构造方法。

【附录】

有根树的计数源程序

```
{$A+,B-,D+,E-,F-,G+,I+,L+,N+,O-,P-,Q-,R-,S+,T-,V+,X+,Y+}
{$M 16384,0,655360}

program Tree1; { 有根树的计数 }
var
  s                       : array [0..29, 1..29] of Comp;
  a                       : array [1..30] of Comp;
  n                       : Integer;
  i, j                    : Integer;

begin
  Write('Please input n(0<n<31): '); Readln(n);
  Fillchar(s, Sizeof(s), 0);
  a[1]:=1;
  for i:=1 to n-1 do begin
    for j:=1 to i do begin
      s[i, j]:=s[i-j, j]+a[i+1-j];
      a[i+1]:=a[i+1]+j*a[j]*s[i, j];
    end;
    a[i+1]:=a[i+1]/i;
  end;
  Writeln(a[n]:1:0);
end.
```

测试数据[②]：

N 的值	程序结果	程序执行时间
1	1	<0.1s
5	9	<0.1s
11	1842	<0.1s
20	12826228	<0.1s
30	354426847597	<0.1s

无根树的计数源程序

```
{$A+,B-,D+,E-,F-,G+,I+,L+,N+,O-,P-,Q-,R+,S+,T-,V+,X+,Y+}
{$M 16384,0,655360}
```

[②] 测试机型：P3-800，256M

```
program Tree1; { 无根树的计数 }
var
  s                      : array [0..29, 1..29] of Comp;
  a                      : array [1..30] of Comp;
  n                      : Integer;
  i, j                   : Integer;
  Res                    : Comp;

begin
  Write('Please input n(0<n<31): '); Readln(n);
  Fillchar(s, Sizeof(s), 0);
  a[1]:=1;
  for i:=1 to n-1 do begin
    for j:=1 to i do begin
      s[i, j]:=s[i-j, j]+a[i+1-j];
      a[i+1]:=a[i+1]+j*a[j]*s[i, j];
    end;
    a[i+1]:=a[i+1]/i;
  end;
  Res:=a[n];
  for i:=1 to n shr 1 do
    Res:=Res-a[i]*a[n-i];
  if not Odd(n) then Res:=Res+(a[n shr 1]+1)*a[n shr 1]/2;
  Writeln(Res:1:0);
end.
```

测试数据：

N 的值	程序结果	程序执行时间
1	1	<0.1s
3	1	<0.1s
10	106	<0.1s
25	104636890	<0.1s
30	14830871802	<0.1s

【参考文献】

《计算机程序设计技巧(The art of computer programming)》第一卷　　D.E.Kunth 著

《组合数学》　　屈婉玲 著