《算法设计与分析》
Fibonacci数

段会川
山东师范大学信息科学与工程学院

2014年12月

Fibonacci数—目录

莱昂纳多·斐波那契(Leonardo Fibonacci)

公元1170 – 1250

意大利数学家；
“中世纪(公元5-15世纪)最有才华的数学家”；
他1202年的著作Liber Abaci(计算之书)向欧洲传播了印度—阿拉伯的十进制计数系统；
“斐波那契数列”以他的名字命名，尽管该数列已被6世纪的印度数学家知晓，但斐波那契是将其引入到欧洲的人。

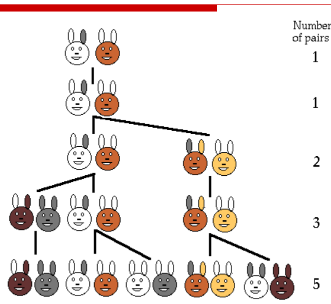
Fibonacci的理想兔模型

- 模型表述
  - 某年1月初有雌雄一对兔，2月初成婚，3月初产一对雌雄仔，以后每一月初均产一对雌雄仔
  - 任何一对雌雄仔均在出生后的第二个月成婚，其后每月产一对雌雄仔
  - 所有兔子均长命不老 n 1年后有多少对兔?
- 数学表达

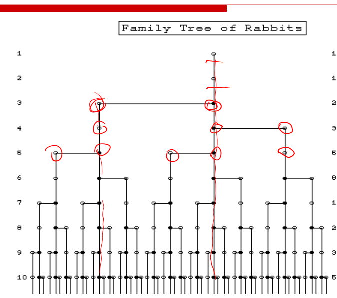
$$F_n = \begin{cases} 1 & n=1 \\ 1 & n=2 \\ F_{n-1}+F_{n-2} & n>2 \end{cases}$$

- 前20个数
  - 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765

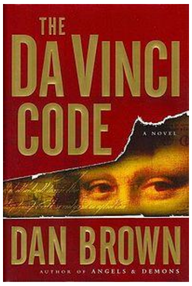
Fibonacci理想兔

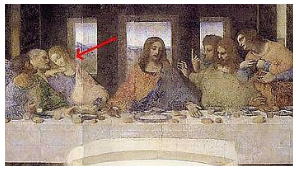


Fibonacci理想兔



小说：达芬奇密码Da Vinci Code (2003)





The Last Supper by Leonardo da Vinci

美国作家丹·布朗(Dan Brown)的一部小说，2003年3月18日由兰登书屋(Random House)出版

电影：达芬奇密码Da Vinci Code (2006)

导演： 朗·霍华德
主演： 汤姆·汉克斯
奥黛丽·塔图
伊安·麦克莱恩
首映： 2006.5.19, 美国

1, 1, 2, 3, 5, 8, 13, …

Fibonacci数求解—递归方法

- 递归方法是计算机科学中普适的求解问题的方法
  - 如果某个较大规模的问题的解依赖于较小规模问题的解，而且较小规模的问题的解决方法与较大规模问题相同，则该问题就可以递归方法求解
  - 但是该过程必须在问题小到一定程度后终结
  - 即问题是自相似、可终结的
- 递归与循环是等价的
  - 大部分程序设计语言都允许函数调用自身而实现递归
  - 函数式程序设计语言(functional programming language)通常不提供命令式程序设计语言(Imperative programming language)中的while和for等控制结构，而仅提供递归机制
- 递归的解法表达通常比循环简单和直观

Fibonacci数求解—递归方法 折半查找

- 简单递归问题示例
  - $n! = n \cdot (n-1)!$
  - $\sum_{i=1}^{n} i = n + \sum_{i=1}^{n-1} i$
  - $x^n = x \cdot x^{n-1}$
  - $\sum_{i=1}^{n} x_i = x_n + \sum_{i=1}^{n-1} x_i$
- 常见的递归算法
  - 二分查找 binary search
  - 树和图的深度和广度优先遍历
  - 文件夹(树型目录系统)和文件的搜索
  - 求最大公约数算法
  - 分治法
  - 汉诺塔问题

Fibonacci数求解—递归算法

```
function FibR(n)
  if n=1 then f = 1
  else if n=2 then f = 1
  else
    f = FibR(n-1)+FibR(n-2)
  return f
end
```

- 递归
  - recursive, recursion
- 调用FibR次数
  - n=1: FibR(1), 共1次 1
  - n=2: FibR(2), 共1次 1
  - n=3: FibR(3) (FibR(1), FibR(2)), 共3次 2
  - n=4: FibR(4) (FibR(3), FibR(2)), 共5次 3
  - n=5: FibR(5) (FibR(4), FibR(3)), 共9次 5

Fibonacci数求解—递归算法

- 调用次数公式

$$T(n) = \begin{cases} 1 & n=1 \\ 1 & n=2 \\ T(n-1)+T(n-2)+1 & n>2 \end{cases}$$

- 调用次数公式计算
  - 1, 1, 3, 5, 9, 15, 24, …
- 对比Fibonacci数
  - 1, 1, 2, 3, 5, 8, 13
- 递归算法的运行次数是一个比Fibonacci数还大的数，即Ω(Fn)，是随n指数式增加的！

算法的伪代码描述

- 伪代码(pseudocode)是算法的描述工具
  - 使用简化的程序设计语言的结构化规范
    如：条件、赋值等成分; if, while, for等控制结构; 基于函数的模块化算法表述; 使用缩格方式表达语句的嵌套关系等
  - 忽略变量定义等对人理解算法不重要的细节
  - 必要时使用简洁的自然语言
  - 必要时使用数学语言
  - 与程序设计语言无关
  - 以凸显算法思想和提高效率为目的

Fibonacci数求解—动态规划算法

```
function FibDP(n)
  for i=1 to n
    if n=1 then f[1] = 1
    else if n=2 then f[2] = 1
    else
      f[i] = f[i-1]+f[i-2]
  endfor
  return f[n]
end
```

space
Θ(n)

- 动态规划
  - Dynamic Programming
  - 保存较小规模的计算结果，使较大规模的计算直接使用较小规模的结果，免去重新计算
- 技巧：使用数组
- 运算次数：Θ(n)，线性的！
- 代价：数组需要耗费存储空间，对巨大的n将因存储空间耗尽而不可行

Fibonacci数求解—简化的动态规划算法

```
function FibDP1(n)
  if n=1 then f=1
  elseif n=2 then f=1
  else
    f1 = 1;  f2 = 1
    for i=3 to n
      f = f1+f2
      f2 = f1; f1 = f
    endfor
  endif
  return f
end
```

F_{n-1} F_{n-2}

Θ(1)

- 技巧：使用两个变量只保存前两个值
- 运算次数：Θ(n)，线性的！
- 完全解决了吗？
  - No：对于巨大的n，Fn将变得超出计算机的整数范围，需要新的方法保存数据，也会带来新的运算开销

幂函数—Θ(logn)算法

```
//计算x^n, Θ(n)算法
function Power(x,n)
  y = 1;
  for i=1 to n
    y = y*x
  endfor
  return y
end
```

- 算法循环n次，线性复杂度Θ(n)

幂函数—Θ(logn)算法

```
//计算x^n , Θ(logn)算法
function PLogN(x,n)
  if n = 1 then y=x
  else
    y0 = PLogN(x, n/2)
    y = y0*y0
    if n is odd then y=y*x
  endif
  return y
end
```

整数除法

- 原理

$$x^n = \begin{cases} x^{n/2} \cdot x^{n/2} & \text{if } n \text{ even} \\ x \cdot x^{\frac{n-1}{2}} \cdot x^{\frac{n-1}{2}} & \text{if } x \text{ odd} \end{cases}$$

公式

  - x^8
    - $x^8 = x^4 \cdot x^4$
    - $x^4 = x^2 \cdot x^2$
    - $x^2 = x \cdot x$
    - 3次乘法！
  - x^{11}
    - $x^{11} = x \cdot x^5 \cdot x^5$
    - $x^5 = x \cdot x^2 \cdot x^2$
    - $x^2 = x \cdot x$
    - 5次乘法！

幂函数—Θ(logn)算法

```
//计算x^n , Θ(logn)算法
function PLogN(x,n)
  if n = 1 then y=x
  else
    y0 = PLogN(x, n/2)
    y = y0*y0
    if n is odd then y=y*x
  endif
  return y
end
```

n/2

k $2^k = n$ $k = \log n$ x^{1024} 10

- 算法递归调用PLogN共logn次
  - n>1时，PLogN执行1次y0*y0，0次或1次y*x
  - 最少的乘法次数是logn次
  - 最多的乘法次数是2(logn)次
  - 具体地说
    - 当n=1024时，执行10次乘法
    - 1024<n<2047时执行11-20次乘法
    - 当n=2048时，执行11次乘法
  - 即算法复杂度为Θ(logn)！

Fibonacci数求解—指数运算公式

- 以矩阵表达Fibonacci数

0 1 1 2 3 5 8

$$\begin{cases} F_{n+1} = 1\cdot F_n + 1\cdot F_{n-1} \\ F_n = 1\cdot F_n + 0\cdot F_{n-1} \end{cases} \Rightarrow \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$$

- 显然这也是一个递推式，因而可有

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

$$F_n = \begin{cases} 1 & n=1 \\ 1 & n=2 \\ F_{n-1}+F_{n-2} & n>2 \end{cases} \qquad F_n = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F_{n-1}+F_{n-2} & n>1 \end{cases}$$

- 将它们合并起来，有

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_n & F_{n-1} \\ F_{n-1} & F_{n-2} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} F_{n-1} & F_{n-2} \\ F_{n-2} & F_{n-3} \end{pmatrix}$$

$$\begin{pmatrix} F_2 & F_1 \\ F_1 & F_0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Fibonacci数求解—指数运算公式

- 设F_0 =0，则Fibonacci数可以从0开始，并且有

$$F_n = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F_{n-1}+F_{n-2} & n\ge 2 \end{cases}$$

- 因此，我们有

$$\begin{pmatrix} F_2 & F_1 \\ F_1 & F_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

- 最后可得关于Fibonacci数的幂表达式

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \qquad x^n$$

Fibonacci数求解—指数运算公式

- 采用与幂函数相同的方式，则有

$$\begin{pmatrix} F_{12} & F_{11} \\ F_{11} & F_{10} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{11} \qquad x^{11}$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{11} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^5 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^5$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^5 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Fibonacci数求解—指数运算公式

- 设计算法时，需要下列式子 x^n $\quad x^{n/2}\cdot x^{n/2}$ $\quad x\cdot(\ )$
  - 假设已经算得矩阵 $\begin{pmatrix} F_a & F_b \\ F_b & F_c \end{pmatrix}$
  - 则 $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} F_a & F_b \\ F_b & F_c \end{pmatrix} = \begin{pmatrix} F_a+F_b & F_b+F_c \\ F_a & F_b \end{pmatrix} = \begin{pmatrix} F_a+F_b & F_a \\ F_a & F_b \end{pmatrix}$

$$\begin{pmatrix} F_a & F_b \\ F_b & F_c \end{pmatrix}\begin{pmatrix} F_a & F_b \\ F_b & F_c \end{pmatrix} = \begin{pmatrix} F_a\cdot F_a+F_b\cdot F_b & F_a\cdot F_b+F_b\cdot F_c \\ F_a\cdot F_b+F_b\cdot F_c & F_b\cdot F_b+F_c\cdot F_c \end{pmatrix}$$

Fibonacci数求解—Θ(logn)算法

```
function FLogN(n)
  if n = 0 then fa=0; fb=-1; fc=-1
  else if n=1 then
    fa=1; fb=0; fc=-1
  else if n=2 then
    fa=1; fb=1; fc=0
  else
    ga,gb,gc = FLogN((n-1)/2)
    fa,fb,fc = Mul2(ga,gb,gc)
    if n-1 is odd then
      fa,fb,fc = Mul1(fa,fb,fc)
    endif
  endif
  return fa,fb,fc
end
```

x^n

```
function Mul2(ga, gb, gc)
  ga2=ga*ga; gb2=gb*gb; gc2=gc*gc
  gab=ga*gb; gbc=gb*gc
  fa = ga2+gb2; fb=gab+gbc
  fc = gb2+gc2
  return fa, fb, fc
end

function Mul1(ga, gb, gc)
  fa = ga+gb;
  fb = ga; fc = gb
  return fa, fb, fc
end
```


Fibonacci数求解—Θ(logn)算法

- 算法递归调用FLogN共log(n-1)次
  - n>2时，FLogN执行1次Mul2，0次或1次Mul1
  - Mul2中包括5次乘法，3次加法
  - Mul1中包括1次加法
  - 算法的乘法次数是5(log(n-1)-1)次
  - 最少的加法次数是3(log(n-1)-1)次
  - 最多的加法次数是4(log(n-1)-1)次
  - 因此，算法复杂度为Θ(logn)!

Fibonacci数的数学解—矩阵特征值法

- $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$
- 令$M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$，则有特征方程$\begin{pmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{pmatrix} = 0$
- 即$(1-\lambda)(-\lambda) - 1 = 0$，$\lambda^2 - \lambda - 1 = 0, \lambda = \frac{1\pm\sqrt{5}}{2}$
- 令$\varphi = \frac{1+\sqrt{5}}{2}, \psi = \frac{1-\sqrt{5}}{2}$
- 则特征向量
  - $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \varphi\begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow \begin{cases} a+b=\varphi a \\ a=\varphi b \end{cases} \Rightarrow V_\varphi = \begin{pmatrix} \varphi \\ 1 \end{pmatrix} b$
  - 同理$V_\psi = \begin{pmatrix} \psi \\ 1 \end{pmatrix} b$

$F_n = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F_{n-1}+F_{n-2} & n>1 \end{cases}$ 同阶的齐次函数

$\lambda - 1 - \frac{1}{\lambda} = 0$　$\lambda - 1 = \frac{1}{\lambda}$

$(\varphi - 1)a = b$　$\frac{1}{\varphi}a = b$　$a = \varphi b$

Fibonacci数的数学解—矩阵特征值法

φ^2　$\varphi^2 a^2 + a^2$　$\psi^2 b^2 + b^2$

- V_φ, V_ψ可以构成正交矩阵$Q = \begin{pmatrix} \varphi a & \psi b \\ a & b \end{pmatrix}$
- 且$Q^{-1} = \begin{pmatrix} \frac{\varphi}{(\varphi^2+1)a} & \frac{1}{(\varphi^2+1)a} \\ \frac{\psi}{(\psi^2+1)b} & \frac{1}{(\psi^2+1)b} \end{pmatrix}$
- 显然$M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = Q\begin{pmatrix} \varphi & 0 \\ 0 & \psi \end{pmatrix}Q^{-1}$
- 因而$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = Q\begin{pmatrix} \varphi^n & 0 \\ 0 & \psi^n \end{pmatrix}Q^{-1}$
- $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} \varphi^{n+1}a & \psi^{n+1}b \\ \varphi^n a & \psi^n b \end{pmatrix}Q^{-1}$

$\varphi\psi ab + ab$　$(\varphi\psi+1)ab$　$\lambda^2 - \lambda - 1 = 0$　韦达定理　$\lambda^2+1 = \lambda+2$

$\varphi\psi = -1$　$\varphi + \psi = 1$

$\frac{\varphi}{\varphi^2+1} + \frac{\varphi\psi}{\psi^2+1}$　$\frac{\varphi}{\varphi+2} + \frac{\psi}{\psi+2}$

$\varphi\psi + 2\varphi + \varphi\psi + 2\psi$　$2\varphi\psi + 2(\varphi+\psi)$

Fibonacci数的数学解—矩阵特征值法

- $Q^{-1} = \begin{pmatrix} \frac{\varphi}{(\varphi^2+1)a} & \frac{1}{(\varphi^2+1)a} \\ \frac{\psi}{(\psi^2+1)b} & \frac{1}{(\psi^2+1)b} \end{pmatrix}$
- $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} \varphi^{n+1}a & \psi^{n+1}b \\ \varphi^n a & \psi^n b \end{pmatrix}Q^{-1}$
- $F_n = \frac{\varphi^{n+1}}{\varphi^2+1} + \frac{\psi^{n+1}}{\psi^2+1} = \frac{\varphi}{\varphi^2+1}\varphi^n + \frac{\psi}{\psi^2+1}\psi^n$
- $\frac{\varphi}{\varphi^2+1} = \frac{\varphi}{\varphi+2} = \frac{1+\sqrt{5}}{5+\sqrt{5}} = \frac{1}{\sqrt{5}}, \frac{\psi}{\psi^2+1} = \frac{\psi}{\psi+2} = \frac{1-\sqrt{5}}{5-\sqrt{5}} = -\frac{1}{\sqrt{5}}$
- 因此$F_n = \frac{1}{\sqrt{5}}\varphi^n - \frac{1}{\sqrt{5}}\psi^n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

$F_n =$

$\varphi = \frac{1+\sqrt{5}}{2}$　$\frac{5+\sqrt{5}}{2}$

$\sqrt{5}(\cancel{\sqrt{5}+1})$

$\psi = \frac{1-\sqrt{5}}{2}$

$\psi + 2 = \frac{5-\sqrt{5}}{2}$

Fibonacci数的数学解—通项公式

- 对于常系数的线性齐次递推式
  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots c_d a_{n-d}$
- 存在一个系数相同的特征多项式
  $p(t) = t^d - c_1 t^{d-1} - c_2 t^{d-2} - \cdots - c_d$
- 该特征多项式的d个(不想等的)根$r_1, r_2, \ldots, r_d$可以构造递推式的闭式解
  $a_n = k_1 r_1^n + k_2 r_2^n + \cdots + k_d r_d^n$
  - 其中，系数$k_1, k_2, \ldots, k_d$可以根据初条件得出

$F_n = F_{n-1} + F_{n-2}$　$a_1 \sim a_d$　$\lambda^2 - \lambda - 1$

Fibonacci数的数学解—通项公式

- Fibonacci数列的递推式为$F_n = F_{n-1} + F_{n-2}$
- 其特征多项式为$p(t) = t^2 - t - 1$
- 该特征多项式的2个根为$\varphi = \frac{1+\sqrt{5}}{2}, \psi = \frac{1-\sqrt{5}}{2}$
- 则Fibonacci数列的通项为$F_n = k_1\varphi^n + k_2\psi^n$
- 由$F_0=0$, $F_1=1$得$\begin{cases} k_1 + k_2 = 0 \\ \frac{1+\sqrt{5}}{2}k_1 + \frac{1-\sqrt{5}}{2}k_2 = 1 \end{cases} \Rightarrow k_1 = \frac{1}{\sqrt{5}}, k_2 = -\frac{1}{\sqrt{5}}$
- 由此得Fibonacci数列的通项公式$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$
- 注意：
  - Fibonacci数列的通项公式对于计算Fibonacci数没有帮助，因为开平方需要复杂的算法，而且用通项公式进行数值计算无法得到整数解。

2.236　$\frac{1.236}{2} < 1$

Fibonacci数—指数式增长

- 显然，$\varphi > 1, -1 < \psi < 0$
- 当$n \to \infty$时，$F_n \to 0.447 \times 2^{0.694n} \approx 2^{0.694n-1}$
- 所以Fn是指数式增长的
- 令0.694n-1=32，得n≈47.55，即第48个Fibonacci数就超过了32位整数的容量
- 令0.694n-1=64，得n≈93.65，即第94个Fibonacci数就超过了64位整数的容量
- 令0.694n-1=50/lg2，得n≈237.91，即第238个Fibonacci数就超过了50位十进制数
- 令0.694n-1=100/lg2，得n≈477.27，即第478个Fibonacci数就超过了100位十进制数

2^n　2^{32}

Fibonacci数—目录

- 算法与数学
  - 理想兔模型
  - 达芬奇密码
  - 递归算法
  - 动态规划算法
  - 幂函数—Θ(logn)算法
  - 指数运算公式
  - Θ(logn)算法
  - 通项公式
  - 指数式增长
- 普适性
  - 攻克希尔伯特10的利器
  - 黄金分割
  - 杨辉三角
  - 黄金螺旋
  - 自然的法则
  - 建筑与艺术设计
  - 光伏阵列设计
  - 五角星

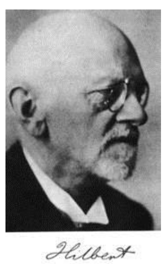
Fibonacci数—希尔伯特的23个问题

- 1900年8月8日，在法国巴黎第二届国际数学家大会上，人类有史以来最伟大的数学家之一德国的大卫·希尔伯特作了题为《数学问题》的演讲，提出了23道最重要的数学问题，其中的许多问题对20世纪数学的发展起到了重要的推动作用

David Hilbert (1862-1943)

Fibonacci数—希尔伯特的第10问题

- 希尔伯特的第10个问题—丢番图方程的可解性，是与计算理论密切相关的
  - 给定一个系数均为有理整数，包含任意个未知数的丢番图方程：
    设计一个过程，通过有限次的计算，能够判定该方程在有理数整数上是否可解



David Hilbert (1862-1943)

Fibonacci数—丢番图与丢番图方程

- 丢番图(Diophantus)
  - 古希腊数学家
    - 公元200(-214)至284(-298)
    - 被誉为代数学的鼻祖
- 丢番图方程 p/q
  Diophantine Equation
  - 不定方程或整系数多项式(polynomial)方程，是变量取值仅容许是整数的整数系数多项式方程

$$p(x_1, x_2, \cdots, x_n) = 0$$

丢番图1621版《算术》

Fibonacci数—丢番图方程示例

- 线性方程：$ax + by = 1$ $ax+by-1=0$ $2x+3y-1=0$
- 费马(Fermat)大定理(费马最后定理)：$x^n + y^n = z^n$
  对于任何的n>2，该不定方程没有非平凡的整数解。
  由17世纪法国数学家费马提出，即“费马猜想”。
  1995年(350年后)英国数学家安德鲁·怀尔斯（Andrew John Wiles）及其学生理查·泰勒（Richard Taylor）最终证明。
- 佩尔(Pell)方程：$x^2 - ny^2 = 1$
- 欧德斯-施特劳斯猜想(Erdős–Straus conjecture)：
  下列方程对任何的n>1都有正整数解，该猜想至今尚未得到证明

$$\frac{4}{n} = \frac{1}{x} + \frac{1}{y} + \frac{1}{z}, \text{即 } 4xyz = n(xy + yz + zx)$$

Fibonacci数—丢番图的墓志铭

- 'Here lies Diophantus,' the wonder behold.
- Through art algebraic, the stone tells how old:
- 'God gave him his boyhood one-sixth of his life,
- One twelfth more as youth while whiskers grew rife;
- And then yet one-seventh ere marriage begun;
- In five years there came a bouncing new son.
- Alas, the dear child of master and sage
- After attaining half the measure of his father's life chill fate took him.
- After consoling his fate by the science of numbers for four years, he ended his life.'

Fibonacci数—丢番图的墓志铭

- 这里安葬着丢番图，
- 下面的代数故事告诉你他的寿命有多长，
- 他的童年占一生的1/6，接着1/12是少年时期，又过了1/7的时光，他找到了终生伴侣，
- 5年之后，婚姻之神赐他一子，
- 可儿子命薄，只活到父亲寿数的一半就匆匆离世，
- 他在失去爱子的伤悲中度过了4年，终于告别科学，撒手人寰。

$$\frac{x}{6}+\frac{x}{12}+\frac{x}{7}+5+\frac{x}{2}+4=x \Rightarrow x=84$$

Fibonacci数—攻克希尔伯特10的利器

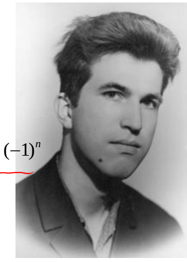
- 希尔伯特第10问题最终由俄罗斯数学家尤里·马蒂雅谢维奇(Yuri Matiyasevich)于1970年攻破，时年他还不到23岁
- 他借用了Fibonacci数如下的性质

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \Rightarrow F_{n+1}F_{n-1}-F_n^2=(-1)^n$$

$$\Rightarrow F_n^2-F_nF_{n-1}-F_{n-1}^2=(-1)^{n+1}$$

$$F_n^2 \mid F_m \Rightarrow F_n \mid m$$

$$\left(\frac{5}{4}\right)^n < F_n < 2^{n-1}$$



Yuri Matiyasevich
March 2, 1947

希尔伯特第10问题—可计算性模型

- 递归函数
  - 库尔特·歌德尔Kurt Gödel)和雅克斯·赫尔布兰德(Jacques Herbrand)，1934
  - Church以及数学家Stephen Kleene和逻辑学家J.B. Rosser一起定义了一类函数，这种函数的值可使用递归方法计算
- λ演算
  - 阿隆佐 邱奇(Alonzo Church)，美国数学家1936年发表可计算函数的第一份精确定义

希尔伯特第10问题—可计算性模型

- 图灵机
  - 阿兰·图灵(Alan Turing)，英国数学家，计算之父《论可计算数在判定问题中的应用(On Computable Numbers, with an Application to the Entscheidungsproblem)》(1936)
- 波斯特机
  - 埃米尔·波斯特(Emil Post)，波兰籍美国数学家和逻辑学家
  - 于1936年独立地提出了与图灵机等价的计算模型(Finite combinatory processes—formulation 1)，称为波斯特机，或图灵—波斯特机
  - 该模型也被证明与递归等同

希尔伯特第10问题—邱奇—图灵论题

- 如果某种方法(算法)可进行运算，那么该运算也可被图灵机执行(也可被递归定义的函数或λ函数执行)
- 任何在算法上可计算的问题同样可由图灵机计算
- 逻辑和数学中的有效或机械方法可由图灵机来表示
  1. 一个方法由有限多简单和精确的指令组成，这些指令可由有限多的符号来描述。
  2. 该方法总会在有限的步骤内产生出一个结果。
  3. 基本上人可以仅用纸张和铅笔来执行。
  4. 该方法的执行不需人类的智慧来理解和执行这些指令。

希尔伯特第10问题—递归可枚举集合

- 可数集合S被称为是递归可枚举、计算可枚举的、半可判定的或可证明的，如果
  - 存在一个算法，只有当输入是S中的元素时，算法才会中止
- 或者等价地说
  - 存在一个算法，可以将S中的成员枚举出来。也就是说该算法的输出就是 S 的成员列表：s1, s2, s3, ... 如果需要它可以永远运行下去。

希尔伯特第10问题—递归可枚举集合

- 根据邱奇-图灵论题可计算函数被图灵机和其他计算模型等价的思想
  - 可数集合 被称为递归可枚举的，如果有一个图灵机，在给定 的一个元素作为输入的时候，总是停机，并在给定的输入不属于 的时候永不停机
- 所有递归集合，即可计算集合，都是递归可枚举的，但不是所有递归可枚举集合都是递归的

希尔伯特第10问题—解决历程

- 1944年，埃米尔·波斯特(Emil Leon Post)首先猜测，对于第10问题，应寻求不可解的证明
- 1949年，马丁·戴维斯(Martin Davis)利用库尔特·哥德尔(Kurt Gödel)的方法，并应用中国余数定理的编码技巧，得到递归可枚举集的戴维斯范式 。
  他注意到丢番图集的补集并非丢番图的。而递归可枚举集对于补集运算也非封闭的，他因此猜测这两个集合类是相同的。

希尔伯特第10问题—解决历程

- 1950年，
  - 朱莉亚·罗宾逊(Julia Robinson)在未知 Davis 工作的情况下，试图证明幂函数$z=y^x$是丢番图的，
  - 虽然并未成功，但是发现如果存在D={(a,b)}使得

    $$(a,b)\in D\Rightarrow b<a^a 且 \forall k>0,\exists(a,b)\in D 使得 b>a^k$$

  - 则幂函数是丢番图的。
  - 该条件简称为J.R.。
  - 并且如果幂函数是丢番图的，那么二项式系数、阶乘以及质数集合都是丢番图的

希尔伯特第10问题—解决历程

- 1959年，David与Putnam 研究了指数丢番图集
  - 指出如果假设存在任意有限长全由质数所组成的算数级数
    - 该假设已于 2004 年由 Ben Green 和 Terence Tao 所证明
  - 则每一个递归可枚举集都是指数丢番图的
  - 因此，若J.R.成立，则可得到结论：
    每一个递归可枚举集都是丢番图的
    因而第十问题是不可解的
- 1960年，Robinson证明了上述的数论假设是不必要的，并且大大简化了证明。
  从而可知，只要能证明幂函数是丢番图的，第十问题就可以解决。
  而关键又是寻找满足 J.R. 假设的丢番图集。

希尔伯特第10问题—解决历程

- 1970年，
  - 尤里·马蒂雅谢维奇指出可由十个一次和二次的联立不定方程组，定义偶角标的斐波那契函数：
    $b=F_{2a}$，其中F_n是第n个斐波那契数。
  - 则它就是丢番图的，并满足 J.R. 假设。
  - 从而可构造出一个不定方程，它不是递归可解的。
    也就是不存在算法，可以计算该方程式的整数解。
- 因此
  - 希尔伯特第十问题最终得到了否定的答案。

希尔伯特第10问题—马蒂雅谢维奇定理

- 满足下列丢番图方程的所有n1,n2,...,nj所构成的j元组集合称为丢番图集

  $$p(n_1,n_2,\cdots,n_j,x_1,x_2,\cdots,x_n)=0$$

- 所有斐波那契数的集是丢番图集，因为若n,x是正整数，则下式成立时，n必是斐波那契数

  $$(n^2-xn-x^2)^2-1=0$$

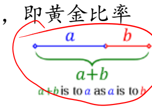
希尔伯特第10问题—马蒂雅谢奇定理

- 每个递归可枚举集都是丢番图集
- 所有丢番图集都是递归可枚举的
- 所有的递归可枚举集都是丢番图的
- 某些递归可枚举集是非递归的，也就是不可计算的
- 因此某些丢番图集是非递归的，也就是不可计算的

Fibonacci数—黄金分割

- 所谓的黄金分割(golden section)是指将线段分成a,b两部分，较长的一段a与较短的一段b的比值等于整个线段a+b与较长的一段a的比值，即黄金比率

$$\frac{a}{b}=\frac{a+b}{a}=\varphi = 1+\frac{1}{\varphi}$$



- 显然φ满足方程：$\varphi^2-\varphi-1=0$
- 其正数解为：$\varphi=\frac{1+\sqrt{5}}{2}\approx 1.6180339887\cdots$

$$\varphi-1=\frac{1}{\varphi} \quad 1.618\cdots-1=\frac{1}{1.618\cdots}$$

Fibonacci数—黄金分割

- 可以看出，Fibonacci数的特征方程正是黄金分割满足的方程，根据韦达定理，我们有

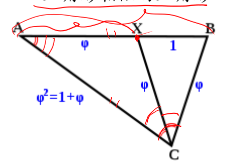
$$\varphi+\psi=1, \varphi\cdot\psi=-1$$

- 相邻Fibonacci数的渐近比为黄金分割 $\lim_{n\to\infty}\frac{F_{n+1}}{F_n}=\varphi$
- 而相邻Fibonacci数的比值是黄金分割最接近的有理数比值：2/1, 3/2, 5/3, 8/5, …
- 黄金分割方程决定了φ很有趣的数值特点

$$\frac{1}{\varphi}=\varphi-1\Rightarrow\frac{1}{1.6180339887\cdots}=0.6180339887\cdots=1.6180339887\cdots-1$$

Fibonacci数—黄金三角形

- 黄金三角形指的是如下性质的等腰三角形
  - 平分其底角得到一个与原三角形相似的三角形



$$\angle BCX=\angle ACX=\angle A$$

$$5\angle A=180^\circ\Rightarrow\angle A=36^\circ$$

$$\frac{BC}{BX}=\frac{AB}{BC}$$

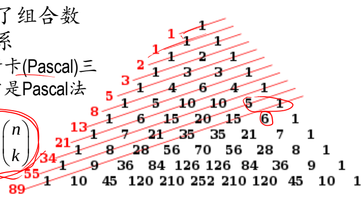
$$BC=CX=AX$$

$$\frac{AX}{BX}=\frac{AB}{AX}=\varphi$$

Fibonacci数—杨辉三角

- 杨辉三角表达了组合数的一种奇特关系
  - 西方称为帕斯卡(Pascal)三角形，因为它是Pascal法则的体现

$$\binom{n-1}{k}+\binom{n-1}{k-1}=\binom{n}{k}$$



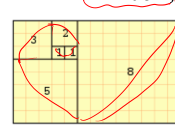
- 杨辉三角各“浅”斜线上数的和正好是Fibonacci数

$$F_n=\sum_{k=0}^{\left\lfloor\frac{n-1}{2}\right\rfloor}\binom{n-k-1}{k}$$

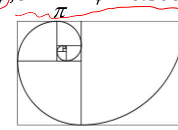
Fibonacci数—黄金螺旋

- 黄金螺旋(golden spiral)指的是增长因子为黄金比率φ的螺旋线，即当角度变化一个直角时，半径扩大到黄金比率φ，

$$r=ae^{b\theta}, e^{b\frac{\pi}{2}}=\varphi, b=\frac{2}{\pi}\ln\varphi\approx 0.306349\cdots$$



使用边长为Fibonacci数的正方形可以长方形方式平铺平面



Fibonacci平铺中依次连接各正方形的相对顶点的圆弧是黄金螺旋非常好的近似

维基百科：http://www.wikipedia.org

Fibonacci数—自然的法则？

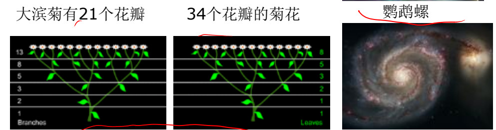
马蹄莲有1个花瓣　大戟属植物有2个花瓣　延龄草有3个花瓣

耧斗菜有5个花瓣　罂粟科植物有8个花瓣　多毛金光菊有13个花瓣

维基百科：*http://www.wikipedia.org*

Fibonacci数—自然的法则？

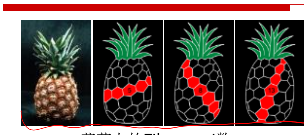
大滨菊有21个花瓣　34个花瓣的菊花　鹦鹉螺



珠蓍的分叉　珠蓍的叶片　螺旋星云

维基百科：*http://www.wikipedia.org*

Fibonacci数—自然的法则？

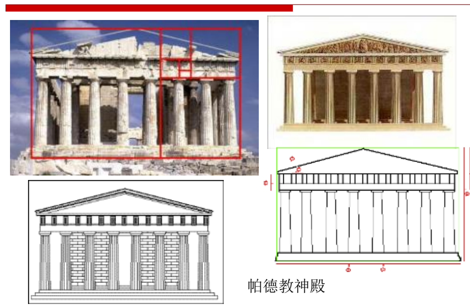


菠萝上的Fibonacci数

金花菊　向日葵仔的排列　向日葵叶子的排列　叶子绕茎的排列

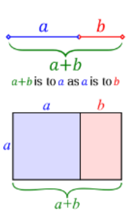
维基百科：*http://www.wikipedia.org*

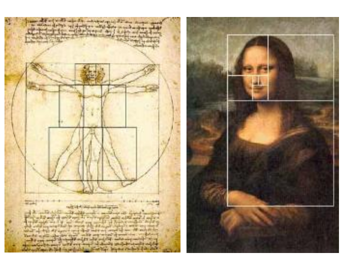
Fibonacci数—建筑设计



帕德教神殿

Fibonacci数—艺术设计

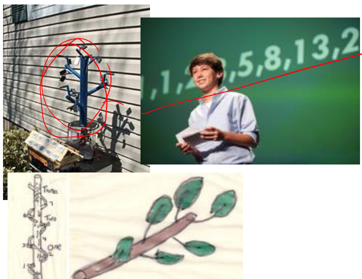




达·芬奇的著名设计

Fibonacci数—光伏阵列设计

Fibonacci数—光伏阵列设计



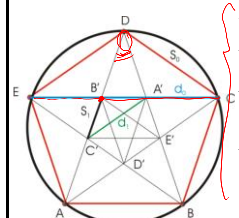
The Secret of the Fibonacci Sequence in Trees

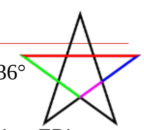
Aidan Dwyer；美国长岛，13岁，小学7年级；获2011年美国自然历史博物馆7-12年级段少年博物学家奖，获奖作品是基于植物叶子沿茎排列的Fibonacci螺旋形态是为了获得充分的光照量的原理设计的光伏阵列

Fibonacci数—红星与希望

Fibonacci数—国家象征的设计

Fibonacci数—五角星





$$\angle ADB = \angle DEC = 36^\circ$$

$$\Delta EA'D \sim \Delta DA'B'$$

$$\frac{EA'}{DB'} = \frac{DB'}{B'A'} \Rightarrow \frac{EA'}{EB'} = \frac{EB'}{B'A'} = \varphi$$

$$\frac{EB'}{EA'} + 1 = \frac{B'A'}{EB'} + 1 \Rightarrow \frac{EC}{EA'} = \frac{EA'}{EB'} = \varphi$$

B'点既是EA'的黄金分割点，也是EC的黄金分割点

五角星丰富地体现了黄金分割

Fibonacci数—总结

- 算法与数学
  - 理想兔模型
  - 达芬奇密码
  - 递归算法
  - 动态规划算法
  - 幂函数—Θ(logn)算法
  - 指数运算公式
  - Θ(logn)算法
  - 通项公式
  - 指数式增长
- 普适性
  - 黄金分割
  - 杨辉三角
  - 黄金螺旋
  - 自然的法则
  - 建筑设计
  - 艺术设计
  - 光伏阵列设计
  - 国家象征的设计
  - 五角星

The End

Thanks