# Chapter 4

# Recurrences (2)

songyou@buaa.edu.cn

# 4.4 The master method（主方法，母函数法）

- solving recurrences of the form

$$T(n) = aT(n/b)+f(n), \tag{4.5}$$

where $a \geq 1$ and $b > 1$, and $f(n)$ is asymptotically positive.

- The master method requires memorization of three cases, but then the solution of many recurrences can be determined quite easily, often without pencil and paper.

（记住三点【其实是一点】，轻松求解）

# From the iteration method to the master theorem

$$T(n)=\begin{cases} c, & n=1 \\ aT\left(\dfrac{n}{b}\right)+cn, & n>1 \end{cases}$$

$$\begin{aligned}
T(n) &= aT(n/b) + cn = a(aT(n/b/b) + cn/b) + cn \\
&= a^2T(n/b^2) + cna/b + cn = a^2T(n/b^2) + cn(a/b + 1) \\
&= a^2(aT(n/b^2/b) + cn/b^2) + cn(a/b + 1) \\
&= a^3T(n/b^3) + cn(a^2/b^2) + cn(a/b + 1) \\
&= a^3T(n/b^3) + cn(a^2/b^2 + a/b + 1) \\
&\ldots \\
&= a^kT(n/b^k) + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1)
\end{aligned}$$

# From the iteration method to the master theorem

$$T(n) = \begin{cases} c, & n = 1 \\ aT\left(\dfrac{n}{b}\right) + cn, & n > 1 \end{cases}$$

$$T(n) = a^k T(n/b^k) + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1)$$

For $n = b^k$, $(k = \log_b n)$,

$$\begin{aligned} T(n) &= a^k T(n/b^k) + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1) \\ &= a^k T(1) + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1) \\ &= ca^k + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1) \\ &= cna^k/b^k + cn(a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1) \\ &= cn(a^k/b^k + a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1) \end{aligned}$$

# From the iteration method to the master theorem

$$T(n)=\begin{cases} c, & n=1 \\ aT\left(\dfrac{n}{b}\right)+cn, & n>1 \end{cases}$$

For $n = b^k$, ($k = \log_b n$),

$T(n) = cn(a^k/b^k + a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1)$

(1) What if $a = b$ ?

$T(n) = cn(k+1) = cn(\log_b n + 1) = \Theta(n\lg n)$

# From the iteration method to the master theorem

$$T(n) = \begin{cases} c, & n = 1 \\ aT\left(\dfrac{n}{b}\right) + cn, & n > 1 \end{cases}$$

For $n = b^k$, ($k = \log_b n$),

$T(n) = cn(a^k/b^k + a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1)$

(1) What if $a = b$ ?

$T(n) = cn(k+1) = cn(\log_b n + 1) = \Theta(n\lg n)$

(2) What if $a < b$ ?

$$\frac{a^k}{b^k} + \frac{a^{k-1}}{b^{k-1}} + \cdots + \frac{a}{b} + 1 = \frac{(a/b)^{k+1} - 1}{(a/b) - 1} = \frac{1 - (a/b)^{k+1}}{1 - (a/b)} < \frac{1}{1 - a/b} = \Theta(1)$$

$T(n) = cn\Theta(1) = \Theta(n)$

# From the iteration method to the master theorem

$$T(n) = \begin{cases} c, & n = 1 \\ aT\left(\frac{n}{b}\right) + cn, & n > 1 \end{cases}$$

For $n = b^k$, ($k = \log_b n$),

$T(n) = cn(a^k/b^k + a^{k-1}/b^{k-1} + a^{k-2}/b^{k-2} + \ldots + a^2/b^2 + a/b + 1)$

(1) What if $a = b$ ?

$T(n) = cn(k+1) = cn(\log_b n + 1) = \Theta(n\lg n)$

(2) What if $a < b$ ?

$T(n) = cn\Theta(1) = \Theta(n)$

(3) What if $a > b$ ?

$$\frac{a^k}{b^k} + \frac{a^{k-1}}{b^{k-1}} + \cdots + \frac{a}{b} + 1 = \frac{(a/b)^{k+1} - 1}{(a/b) - 1} = \Theta\left((a/b)^k\right)$$

$$T(n) = cn\cdot\Theta\left((a/b)^k\right) = cn\cdot\Theta\left(\frac{a^k}{b^k}\right) = cn\cdot\Theta\left(\frac{a^{\log_b n}}{n}\right)$$

$$= \Theta\left(a^{\log_b n}\right) = \Theta\left(n^{\log_b a}\right)$$

# From the iteration method to the master theorem

$$T(n)=\begin{cases} c, & n=1 \\ aT\left(\dfrac{n}{b}\right)+cn, & n>1 \end{cases}$$

$$T(n)=\begin{cases} \Theta(n) & , \quad a<b \quad (n^{\log_b a}<n) \\ \Theta(n\log_b n) & , \quad a=b \quad (n^{\log_b a}=n) \\ \Theta\left(n^{\log_b a}\right) & , \quad a>b \quad (n^{\log_b a}>n) \end{cases}$$

# 4.4.1 The master theorem

- **Theorem 4.1**

  Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

  $$T(n) = aT(n/b) + f(n),$$

  where we interpret $n/b$ to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows.

1. If $f(n) = O(n^{(\log_b a)-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{(\log_b a)+\varepsilon})$ for some constant $\varepsilon > 0$,

   and if $\underline{af(n/b) \leq cf(n)}$ for some constant $c < 1$

   and all sufficiently large $n$, then $T(n) = \Theta(f(n))$.

_____的意思是 $a*f(n/b) > f(n)$ 不能得出 $T(n) = d*f(n)$

Idea of proof: Let $T(n) = d*f(n)$, then $T(n) = a*T(n/b) + f(n) \rightarrow d*f(n) = a*d*f(n/b) + f(n)$

If $a.\ f(n/b) > f(n)$, then $d*f(n) = a*d*f(n/b) + f(n) > d*f(n)+f(n) = (d+1)*f(n) \rightarrow d > d+1$

# 4.4.1 The master theorem: understand what it says

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \Bigg\} \exists \varepsilon > 0, c < 1$$

Comparing the function $f(n)$ with $n^{\log_b a}$. Intuitively, the solution is determined by the larger of the two functions.

- Case 1, $n^{\log_b a}$ larger, then the solution is $T(n) = \Theta(n^{\log_b a})$.
- Case 3, $f(n)$ larger, then the solution is $T(n) = \Theta(f(n))$
- Case 2, the two functions are the same size, we multiply by a logarithmic factor, and the solution is $T(n) = \Theta(n^{\log_b a}\lg n) = \Theta(f(n)\lg n)$ .

Case 1: $f(n) \le n^{(\log_b a)-\varepsilon} = n^{(\log_b a)} / n^{\varepsilon} < n^{(\log_b a)}$

# 4.4.1 The master theorem: special cases

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a} \lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \Bigg\} \exists \varepsilon > 0, c < 1$$

- Polynomially
  - Case 1, $f(n)$ must be polynomially smaller than $n^{\log_b a}$.
  - Case 3, $f(n)$ must be polynomially larger than $n^{\log_b a}$.
- Gap

  Example: $f(n) = n \lg n$, $n^{\log_b a} = n$

  - There is a gap between cases 1 and 2 when $f(n)$ is smaller than $n^{\log_b a}$ but not polynomially smaller.
  - Similarly, there is a gap between cases 2 and 3 when $f(n)$ is larger than $n^{\log_b a}$ but not polynomially larger.

# 4.4.2 Using the master method: some examples

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta\left(f(n)\right), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \left.\vphantom{\begin{cases}a\\b\\c\end{cases}}\right\} \begin{matrix} \exists \varepsilon > 0 \\ , c < 1 \end{matrix}$$

- $T(n) = 9T(n/3) + n$

- $T(n) = T(2n/3) + 1$

## 4.4.2 Using the master method: some examples

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \Bigg\} \begin{matrix} \exists \varepsilon > 0 \\ , c < 1 \end{matrix}$$

- $T(n) = 9T(n/3) + n$

$$a = 9, b = 3, f(n) = n \quad \Rightarrow \quad n^{\log_b a} = n^{\log_3 9} = n^2 = \Theta(n^2)$$

$$\Rightarrow \quad f(n) = O(n^{\log_3 9 - \varepsilon}), \text{where } \varepsilon = 1 \quad \Rightarrow \quad T(n) = \Theta(n^2)$$

- $T(n) = T(2n/3) + 1$

$$a = 1, b = 3/2, f(n) = 1 \quad \Rightarrow \quad n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$$

$$\Rightarrow \quad f(n) = \Theta(n^{\log_b a}) \quad \Rightarrow \quad T(n) = \Theta(\lg n)$$

## 4.4.2 Using the master method: some examples

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \Bigg\} \exists \varepsilon > 0, c < 1$$

- $T(n) = 3T(n/4) + n\lg n$

$$a = 3, b = 4, f(n) = n\lg n \quad \Rightarrow \quad n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$$

$$\Rightarrow \quad f(n) = \Omega(n^{(\log_4 3)+\varepsilon}), \text{where } \varepsilon \approx 0.2, \text{ and for sufficiently large } n,$$

$$af(n/b) = 3(n/4)\lg(n/4) \le (3/4)n\lg n = cf(n) \text{ for } c = 3/4$$

$$\Rightarrow \quad T(n) = \Theta(n\lg n)$$

# 4.4.2 Using the master method: some examples

$$T(n) = aT(n/b) + f(n),$$

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right), & f(n) = O\left(n^{(\log_b a)-\varepsilon}\right) \\ \Theta\left(n^{\log_b a}\lg n\right), & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta(f(n)), & f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) \text{ and } af(n/b) \le cf(n) \text{ for large } n \end{cases} \left.\vphantom{\begin{matrix}a\\b\\c\end{matrix}}\right\} \begin{matrix} \exists\varepsilon > 0 \\ ,c < 1 \end{matrix}$$

- $T(n) = 2T(n/2) + n\lg n$

$$a = 2, b = 2, f(n) = n\lg n \quad \Rightarrow \quad n^{\log_b a} = n^{\log_2 2} = n^1 = n,$$

but $f(n)/n = \lg n$, which is asymptotically less than $n^{\varepsilon}$ for any positive constant $c$, that is $f(n)$ is not polynomially larger than $n$. Consequently, the recurrence falls into the gap between case 2 and case 3.

$f(n) = n\lg n \ge n^{\log_b a} = n^{\log_2 2} = n$，因此，只可能是情况3。情况3中要求为多项式大，即存在 $\varepsilon$，使得 $f(n)/n^{\log_b a} \ge n^{\varepsilon}$。但 $n\lg n / n = \lg n < n^{\varepsilon}$，对任意小的 $\varepsilon$ 均成立，矛盾。此时，介于情况2和3之间，也就是主定理不适用。

# Exercises and problems

**The substitution method is powerful, but it can be applied only in cases when it is easy to guess the form of the answer.**

## E Zexal的二叉树（签到）

时间限制：1000ms 内存限制：65536kb

通过率：200/209 (95.69%) 正确率：200/596 (33.56%)

### 题目

知识点：树，数论，dp，递归（都可以做）

上学期我们学习了二叉树，也都知道3个结点的二叉树有5种， 现给你二叉树的结点个数$n$，要你输出不同形态二叉树的种数。

### 输入

第一个数为一个整数$n(n <= 30)$

### 输出

对于每组数据，输出一行，不同形态二叉树的种数。

### 输入样例

```
3
```

### 输出样例

```
5
```

## recursion

$h(n) = h(0)*h(n\text{-}1) + h(1)*h(n\text{-}2) + \ldots + h(n\text{-}1)*h(0)$

**另一种递归式(其中 $n >= 2$) :**

$h(n) = ((4*n\text{-}2)/(n+1))*h(n\text{-}1)$

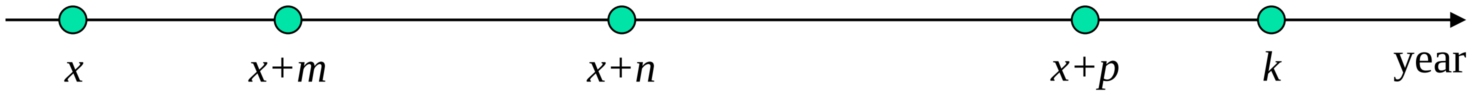
这个就不好猜!

$$\boldsymbol{h(n) = C(2n, n)/(n+1)}$$
$$\boldsymbol{(n = 1, 2, 3, \ldots)}$$

习题：证明该公式。

# Exercises and problems

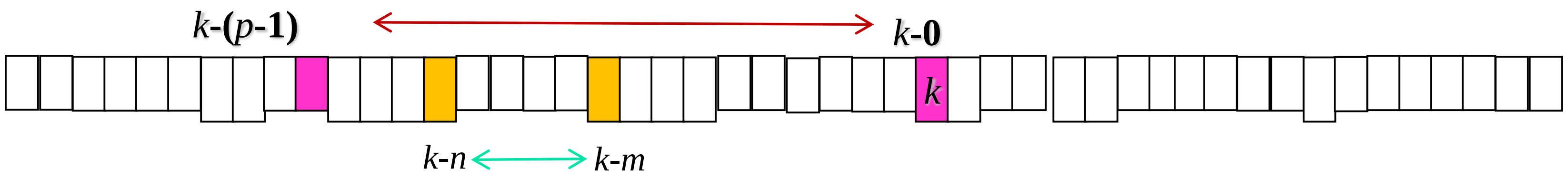
一头 $x$ 年出生的母牛从 $x+m$ 年到 $x+n$ 年间每年生出一头母牛，并在 $x+p$ 年被淘汰。写一个程序，按顺序读入整数 $m, n, p, k$ $(3<m<n<p<60, 0<k<60)$，设第 0 年有一头刚出生的母牛，计算第 $k$ 年时共存有多少头未被淘汰的母牛。（母牛会老也会死的情况）




**题目分析：**

- 第 $k$ 年母牛的**总数量** $T(k)$ 为第 $k$-($p$-1) 年到第 $k$ 年新出生母牛数量之和（不超过$p$岁）（超过$p$岁的，即$k$-$p$年及以前出生的在第$k$年时都死了）。

$$T(k) = N(k-p+1) + N(k-p) + \ldots + N(k-0) \quad \text{-----------} \quad (1)$$




- 第 $k$ 年**新生母牛** $N(k)$ 等于 $k$-$m$ 年到 $k$-$n$ 年出生母牛数量之和（即这期间出生的母牛在第$k$年有生产能力）（$k$-$n$年前出生的牛太老了，不能再生产，$k$-$m$年后出生的牛太小，还不能生产）

$$N(k) = N(k-n) + N(k-n+1) + \ldots + N(k-m) \quad \text{-----------} \quad (2)$$

基于(1)和(2)，可写递归程序。
基本情况？（请读者补充）

$T(n)$的时间复杂度分析？

# Exercises and problems

## Fibonacci 数列 ($F_n = F_{n-1} + F_{n-2}$) 的母牛问题?

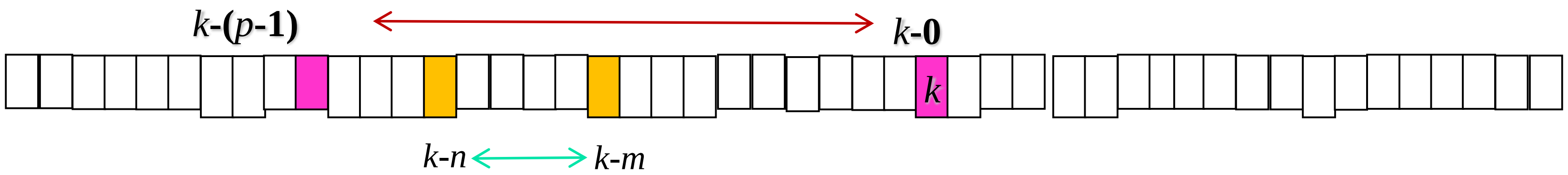
第 1 年（2001年）有一头新生母牛，两岁开始（2003年起）生产，每年生产一头母牛，第 $k$ 年共有多少头母牛?

传统的母牛问题（上一页的问题）：

- 第 $k$ 年母牛的总数量$T(k)$为第 $k$-($p$-1) 年到第 $k$ 年新出生母牛数量之和（不超过$p$岁）（超过 $p$ 岁的，即 $k$-$p$ 年及以前出生的在第 $k$ 年时都死了）。

$T(k) = N(k-p+1) + N(k-p) + \ldots + N(k-0)$ --- (1)

**如果母牛不死:**
$T(k) = N(1) + \ldots + N(k-1) + N(k-0)$




- 第 $k$ 年新生母牛 $N(k)$ 等于 $k$-$m$ 年到 $k$-$n$ 年出生母牛数量之和（即这期间出生的母牛在第$k$年有生产能力）（$k$-$n$ 年前出生的牛太老了，不能再生产，$k$-$m$ 年后出生的牛太小，还不能生产）

$N(k) = N(k-n) + N(k-n+1) + \ldots + N(k-m)$ --- (2)

**如果母牛不老（一直能生）：**
$N(k) = N(1) + N(2) + \ldots + N(k-2)$

由(1)得 $T(k) = T(k-1) + N(k)$
由(2)得 $N(k) = T(k-2)$
综上，$T(k) = T(k-1) + T(k-2)$

可见，Fib数是母牛问题的特例！

# Exercises and problems

## 思考题：

1. 一般母牛问题：母牛要老，要死（前两页）
2. Fib 母牛问题：母牛不老，不死（前一页）
3. 特别母牛问题：母牛不死，要老，如何求解？