《算法设计与分析》讲义

2019-2020（2）

1. 概述 （第 1 讲）

1.1. 背景

- 从实际问题谈起
- **算法**

(1) 解决问题：正确性

(2) 解决效率：复杂度

1.2. 计算机科学与技术难度级别（本课程难度级别：2.5）

- 技术

(1) 程序设计：语言学习；实现算法的工具

(2) 数据结构：语言应用；实现算法的基础

(3) 编译原理：语言翻译；算法的高难应用

- 科学

(4) 计算理论：算法边界；可计算和不可计算

(5) 数理逻辑：数学边界：可证明和不可证明

1.3. 算法难度级别

(1) 某个算法：二分查找、快速排序
  - 设计
  - 分析**（某个算法）复杂度**

(2) 某个问题：检索问题、排序问题
  - 设计算法类：同一个问题的多个算法
  - 分析**（某个问题）复杂度**

(3) 不同问题
  - 设计不同的算法类
  - 分析**（不同问题）的难度**
    - P 问题：多项式可验证
    - NP 问题：一般 NP 问题；NP 完全（NP-Complete）问题（NP 问题和 NP 难问题的交集）
    - NP 难（NP-hard）问题：多项式可能无法验证

1.4. 课程主要内容

(1) 算法的数学基础

(2) 典型算法的设计与分析，如分治策略、动态规划、贪心法、回溯、分支限界等

(3) 典型问题的分析：检索问题、排序问题、选择问题等

1.5. 课程前后关系

(1) 基础课程：《程序设计》、《数据结构》

(2) 后续课程：《机器学习》、《编译原理》、《计算理论》

1.6. 课程作业与考核

(1) 《算法分析与设计》**周二 6、7：13:20--14:50**
  - 平时作业 50%：课堂纪律和出勤 10%；5 次平时作业 40%
  - 卷面考试 50%

(2) 《算法分析与设计实践》：**周二 10、11：18:00—19:30**

- 平时作业 50%：课堂纪律和出勤 10%；8 次平时作业 40%
- 大作业 50%：例如，“异序作业排序与调度”的两道大题

(3) 作业形式

- Blog：作业对应文档均已博客形式维护
- Github：作业代码均在 git 上维护
- Note（读书笔记）：阅读课外算法相关书籍，完成读后感一篇

1.7. 数学与算法课的关系

(1) 问题的数学建模：形式化 & 半形式化 & 非形式化

(2) 算法的正确性证明：好-算法正确

(3) 算法的复杂度计算：好-算法效率高

2. 函数的渐进的界（第 2 讲）

定义 1.1 设 f 和 g 是定义域为自然数集 N 上的函数

(1) 若存在正数c和n_0，使得对于一切$n > n_0$有$0 \le f(n) \le cg(n)$成立，则称$f(n)$的**渐进的上界**是$g(n)$，记作$f(n) = O(g(n))$；$\exists c, n_0$

(2) 若存在正数c和n_0，使得对于一切$n > n_0$有$0 \le cg(n) \le f(n)$成立，则称$f(n)$的**渐进的下界**是$g(n)$，记作$f(n) = \Omega(g(n))$；

(3) 若对于任意正数c都存在n_0，使得对于一切$n > n_0$有$0 \le f(n) < cg(n)$成立，则记作$f(n) = o(g(n))$；$\forall c, \exists n_0$

(4) 若对于任意正数c都存在n_0，使得对于一切$n > n_0$有$0 \le cg(n) < f(n)$成立，则记作$f(n) = \omega(g(n))$；

(5) 若$f(n) = O(g(n))$且$f(n) = \Omega(g(n))$，则记作$f(n) = \Theta(g(n))$，$g(n)$是$f(n)$的**紧的界**。□

注：

(1) 定义域为自然数：考虑到算法的实际应用场景，n 表示问题的规模，f(n)表示时间复杂度；

(2) 考虑算法性能：关心实例规模很大的时候算法所表现的计算效率；

(3) 算法的时间复杂度函数$T(n)$的阶：影响算法性能的关键因素，n 为问题规模；

(4) 希腊字母读音

- OoOmicorn /əuˈmaikrən/ 或 /ˈɑmɪˌkrɑn/
- ΩωOmega
- ΘθTheta

(5) $f(n) = O(g(n))$，$f(n)$的阶可能低于$g(n)$阶，也可能等于

(6) $f(n) = o(g(n))$，$f(n)$的阶只可能低于$g(n)$阶

(7) $f(n) = n^2 + n$

- $f(n) = O(n^2)$
- $f(n) = O(n^3)$
- $f(n) = o(n^3)$

3. 函数的渐进的界相关定理

3.1. 定理 1.1

设 f 和 g 是定义为自然数集合 N 上的非负函数

(1) 如果$\lim_{n\to\infty}\frac{f(n)}{g(n)}$存在，并且等于某个常数 c>0，那么$f(n) = \Theta(g(n))$

(2) 如果$\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$，那么$f(n) = o(g(n))$

(3)　如果$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=+\infty$，那么$f(n)=\omega(g(n))$

3.1.1. 定理 1.1（1）证明

证明：假设$f(n)$和$g(n)$均大于等于 0

如果$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}$存在，并且等于某个常数 c>0，即$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=c>0$。

对于$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=c$，根据极限定义，任意给定正数$\varepsilon$，总存在$n_0$使得对于$n>n_0$的一切$n$，对应的$\frac{f(n)}{g(n)}$都满足$|\frac{f(n)}{g(n)}-c|<\varepsilon$。

于是，我们给出下面的**推导**：

$|\frac{f(n)}{g(n)}-c|<\varepsilon$

$\Rightarrow|\frac{f(n)}{g(n)}-c|<\varepsilon$

$\Rightarrow c-\varepsilon<\frac{f(n)}{g(n)}<c+\varepsilon$ 任意给定正数ε，则取$\varepsilon=c/2$

$\Rightarrow\ c/2<\frac{f(n)}{g(n)}<3c/2<2c$

$\Rightarrow c/2\le\frac{f(n)}{g(n)}<3c/2\le 2c$

$\Rightarrow\frac{c}{2}g(n)\le f(n)\le 2cg(n)$

因此，

- 存在c和n_0，使得对于一切$n>n_0$有$0\le f(n)\le cg(n)$成立，则根据函数的渐进的上界的定义，$f(n)$的渐进的上界是$g(n)$，即$f(n)=O(g(n))$；
  - ∃c：$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=c>0$
  - $\exists n_0$：极限定义，对于$\forall\varepsilon$，$\exists n_0$使得对于$n>n_0$的一切n
- 存在c和n_0，使得对于一切$n>n_0$有$0\le cg(n)\le f(n)$成立，则根据函数的渐进的下界的定义，$f(n)$的渐进的下界是$g(n)$，即$f(n)=\Omega(g(n))$；
- 根据函数的渐进的界的定义，$f(n)=O(g(n))$且$f(n)=\Omega(g(n))$，即$f(n)=\Theta(g(n))$。
  □

3.1.2. 定理 1.1（2）证明

证明：假设$f(n)$和$g(n)$均大于等于 0

由于$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=0$，根据极限定义，任意给定正数$\varepsilon$，总存在$n_0$使得对于$n>n_0$的一切$n$，对应的$\frac{f(n)}{g(n)}$都满足$|\frac{f(n)}{g(n)}-0|<\varepsilon$，则

$|\frac{f(n)}{g(n)}-0|<\varepsilon$

$\Rightarrow|f(n)|<\varepsilon g(n)$

因此，对于任意正数ε都存在n_0，使得对于一切$n>n_0$有$0\le f(n)<\varepsilon g(n)$成立，即$f(n)=$

$o(g(n))$；

3.1.3. 定理 1.1（3）证明

证明：假设$f(n)$和$g(n)$均大于等于 0

由于$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=+\infty$，根据极限定义，任意给定正数M，总存在$n_0$使得对于$n>n_0$的一切$n$，对应的$\frac{f(n)}{g(n)}$都满足$\left|\frac{f(n)}{g(n)}\right|>\mathrm{M}$，则

$$\left|\frac{f(n)}{g(n)}\right|>\mathrm{M}$$

$\Rightarrow |f(n)|>\mathrm{M}g(n)$

因此，对于任意正数M都存在n_0，使得对于一切$n>n_0$有$0\le \mathrm{M}g(n)<f(n)$成立，即$f(n)=\omega(g(n))$。

3.2. 定理 1.2

设 *f, g, h* 是定义域为自然数集合的函数

（1） 如果$f=O(g)$且$g=O(h)$，那么$f=O(h)$

（2） 如果$f=\Omega(g)$且$g=\Omega(h)$，那么$f=\Omega(h)$

（3） 如果$f=\Theta(g)$且$g=\Theta(h)$，那么$f=\Theta(h)$

定理 1.2（1）

证明：

由于$f=O(g)$，那么**根据渐进的上界定义**，存在正数c_1和n_1，使得对于一切$n>n_1$有$0\le f(n)\le c_1g(n)$成立；

由于$g=O(h)$，那么**根据渐进的上界定义**，存在正数c_2和n_2，使得对于一切$n>n_2$有$0\le g(n)\le c_2h(n)$成立；

由此，

$0\le f(n)\le c_1g(n)$

$\Rightarrow 0\le f(n)\le c_1g(n)\le c_1c_2h(n)$，*n* 必须同时满足$n>n_1$和$n>n_2$。取$n_0=\max(n_1,n_2)$，即$n>n_0$。

因此，存在正数$\mathrm{c}=c_1c_2$和$n_0=\max(n_1,n_2)$，使得对于一切$n>n_0$有$0\le f(n)\le ch(n)$成立，则 $f(n)=O(h(n))$。

3.3. 关于几类函数的阶的结果

- **多项式函数**：$f(n)=a_0+a_1n+a_2n^2+\cdots+a_dn^d$，有$f(n)=\Theta(n^d)$
- **对数函数**：对每个 b>1 和每个 a>0，有$\log_b n=o(n^a)$；对不同的 a 和 b，a，b>0，有$\log_b n=\Theta(\log_a n)$
- **指数函数**：对每个 r>1 和每个 d>0，r^n满足$n^d=o(r^n)$
- **阶乘函数**：$n!=\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)$

注:

- **对数函数<多项式函数<指数函数<阶乘函数**

  (1) 证明（对数函数）：对每个 b>1 和每个 a>0，有$\log_b n=o(n^a)$

$$\lim_{n\to\infty}\frac{\log_b n}{n^a}$$
$$=\lim_{n\to\infty}\frac{lnn}{lnb*n^a}$$
$$=\lim_{n\to\infty}\frac{(lnn)^{'}}{(lnb*n^a)^{'}}$$
$$=\lim_{n\to\infty}\frac{\frac{1}{n}}{lnb*an^{a-1}}$$
$$=\lim_{n\to\infty}\frac{1}{lnb*an^a}$$
$$=0$$

(2) 证明（对数函数）：对不同的 a 和 b，a, b>0，有$\log_b n=\Theta(\log_a n)$

$$\lim_{n\to\infty}\frac{\log_b n}{\log_a n}$$
$$=\lim_{n\to\infty}\frac{\log_b n}{\log_a n}$$
$$=\lim_{n\to\infty}\frac{\lg n*\lg a}{\lg n*\lg b}$$
$$=\log_b a$$

(3) 证明：对每个 r>1 和每个 d>0，r^n满足$n^d=o(r^n)$

若 d<=1，显然有

$$\lim_{n\to\infty}\frac{n^d}{r^n}$$
$$=\lim_{n\to\infty}\frac{dn^{d-1}}{r^n lnr}$$
$$=\frac{d}{lnr}\lim_{n\to\infty}\frac{n^{d-1}}{r^n}$$
$$=\frac{d}{lnr}\lim_{n\to\infty}\frac{1}{n^{1-d}r^n}$$
$$=0$$

若 d>1，根据洛必达法则，n^d将通过逐次求导降低为n^{d-1}，n^{d-2}，…,直到n^{d_0},其中$d_0<1$,而分母的指数函数不变，根据前面的结果，上式极限为 0。

(4) 证明：阶乘函数渐进的界$\log(n!)=\Theta(n\log n)$

$$\lim_{n\to\infty}\frac{\log(n!)}{n\log n}$$
$$=\lim_{n\to\infty}\frac{\ln(n!)/\ln 2}{n\ln n/\ln 2}$$

$$= \lim_{n\to\infty} \frac{\ln(n!)}{n\ln n}$$

$$= \lim_{n\to\infty} \frac{\ln\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n (1+(\frac{c}{n})\right)}{n\ln n}, \text{ by } Stirling\ formula, n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1+\Theta\left(\frac{1}{n}\right)\right)$$

$$= \lim_{n\to\infty} \frac{\ln\left(\sqrt{2\pi n}\right) + \ln\left(\frac{n}{e}\right)^n + \ln\left(1+(\frac{c}{n})\right)}{n\ln n}$$

$$= \lim_{n\to\infty} \frac{(\frac{1}{2}\ln 2\pi + \frac{1}{2}\ln n) + (n\ln n - n\ln e) + (\ln(n+c) - \ln n)}{n\ln n}$$

$$= \lim_{n\to\infty} \frac{n\ln n + \ln(n+c)}{n\ln n}$$

$$= \lim_{n\to\infty} 1 + \lim_{n\to\infty} \frac{\ln(n+c) - \ln n + \ln n}{n\ln n}$$

$$= \lim_{n\to\infty} 1 + \lim_{n\to\infty} \frac{\ln(n+c) - \ln n}{n\ln n}$$

$$= \lim_{n\to\infty} 1 + \lim_{n\to\infty} \frac{\ln(1+\frac{c}{n})}{n\ln n}$$

$$= \lim_{n\to\infty} 1 + \lim_{n\to\infty} \frac{0}{\infty\cdot\infty}$$

$$= \lim_{n\to\infty} 1$$

$$= 1$$

(5) $r^n = o(n!)$

$$\lim_{n\to\infty} \frac{r^n}{n!}$$

$$= \lim_{n\to\infty} \frac{r\cdot r\cdots r}{1\cdot 2\cdots r} \cdot \frac{r\cdots r}{(r+1)\cdots n}$$

$$= \lim_{n\to\infty} C \cdot \frac{r\cdots r}{(r+1)\cdots n}$$

$$\leq \lim_{n\to\infty} C\cdot\frac{r}{n}$$

$$= 0$$

4. 求和的方法（第 3 讲）

4.1. 基本的求和公式

(1) 等差级数：可用数学归纳法证明

$$\sum_{k=1}^{n} a_k = \frac{n(a_1+a_n)}{2}$$

(2) 等比级数：可用数学归纳法证明

$$\sum_{k=0}^{n} aq^k = \frac{a(1-q^{n+1})}{1-q}$$

$$\sum_{k=1}^{n} aq^{k-1} = \frac{a(1-q^n)}{1-q}$$

(3) 调和级数：(近似值，后面给出证明)

$$\sum_{k=1}^{n} \frac{1}{k} = lnn + O(1)$$

4.2. 估计和式上界

- 估计和式上界：放大法（最大项代替序列的每一项目）

$$\sum_{k=1}^{n} a_k \le na_{max}$$

- 估计和式上界：放大法（用到等比级数）。

  假设存在常数 r<1，使得$\frac{a_{k+1}}{a_k} \le r$对于一切 k>=0 成立，那么有：

$$\sum_{k=1}^{n} a_k \le \sum_{k=1}^{\infty} a_0 r^k = a_0 \sum_{k=0}^{\infty} r^k = a_0 \lim_{n\to\infty} \frac{1-r^{n+1}}{1-\mathrm{r}} = \frac{a_0}{1-r}$$

- 估计和式上界：积分法

  例 1.9 估计$\sum_{k=1}^{n} \frac{1}{k}$的渐进的界。

  解：

(1) 下界$\sum_{k=1}^{n} \frac{1}{k} \ge \int_1^{\mathrm{n}+1} \frac{dx}{x} = \ln{(n+1)}$，即 1,1/2,1/3 的下界是$\int_1^{\mathrm{n}+1} \frac{dx}{x}$

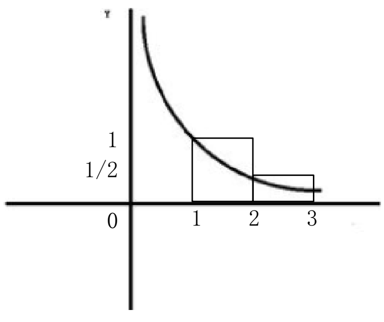


图 1 下界

(2) 上界$\sum_{k=1}^{n} \frac{1}{k} = 1 + \sum_{k=2}^{n} \frac{1}{k} \le 1 + \int_1^{\mathrm{n}} \frac{dx}{x} = \ln(n) + 1$，即 1/2,1/3 的上界是$\int_1^{\mathrm{n}} \frac{dx}{x}$

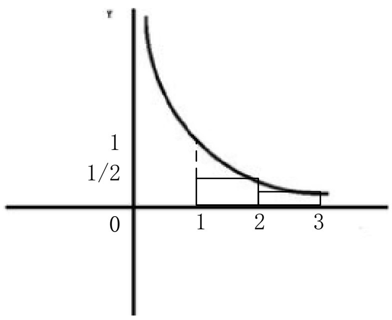


图 2 上界

(3) $\sum_{k=1}^{n}\frac{1}{k}=\Theta(\text{logn})$

5. 递推方程的求解方法

5.1. 迭代归纳法

用迭代归纳法求解以下递推方程

$$\begin{cases} W(n)=2W\left(\frac{n}{2}\right)+n-1, n=2^k \\ W(1)=0 \end{cases}$$

$W(n)$

$=2W\left(\frac{n}{2}\right)+n-1$

$=2W\left(\frac{2^k}{2}\right)+2^k-1$

$=2W(2^{k-1})+2^k-1$

即，$W(2^k)=2W(2^{k-1})+2^k-1$

$W(n)$

$=2W(2^{k-1})+2^k-1$ 代入

$=2(2W(2^{k-2})+2^{k-1}-1)+2^k-1$ 展开

$=(2^2W(2^{k-2})+2^k-2)+2^k-1$ 整理

$=2^2W(2^{k-2})+(2^k-2)+(2^k-1)$ 代入

$=2^2(2W(2^{k-3})+2^{k-2}-1)+(2^k-2)+(2^k-1)$ 展开

$=2^3W(2^{k-3})+(2^k-2^2)+(2^k-2)+(2^k-1)$ 整理

$=\cdots$

$=2^kW(2^{k-k})+(2^k-2^{k-1})+\cdots+(2^k-2)+(2^k-1)$

$=2^kW(1)+k2^k-(2^{k-1}+2^{k-2}+\cdots+2+1)$

$=2^kW(1)+k2^k-\frac{1-2^k}{1-2}$

$=k2^k-2^k+1$

$=nlogn-n+1$，这里$logn=\log_2 n$

证明：$W(n)=nlogn-n+1$，$n=2^k$

思路：数学归纳法，（1）证明 n=1 成立（2）假设$n=2^{k-1}$成立时，证明$n=2^k$亦成立

1) n=1，$W(1) = 1log1 - 1 + 1 = 0$

2) 假设$n = 2^{k-1}$成立，即$W(2^{k-1}) = 2^{k-1}log2^{k-1} - 2^{k-1} + 1$

$W(n)$

$= 2W\left(\frac{n}{2}\right) + n - 1$

$= 2W(2^{k-1}) + 2^k - 1$，由归纳假设

$= 2(2^{k-1}log2^{k-1} - 2^{k-1} + 1) + 2^k - 1$

$= 2 * 2^{k-1}log2^{k-1} - 2^k + 2 + 2^k - 1$

$= 2^k(k-1)log2 - 2^k + 2 + 2^k - 1$

$= 2^k(k-1) - 2^k + 2 + 2^k - 1$

$= k2^k - 2^k - 2^k + 2 + 2^k - 1$

$= k2^k - 2^k + 1$

$= nlogn - n + 1$，Q.E.D.

5.2. 递归树

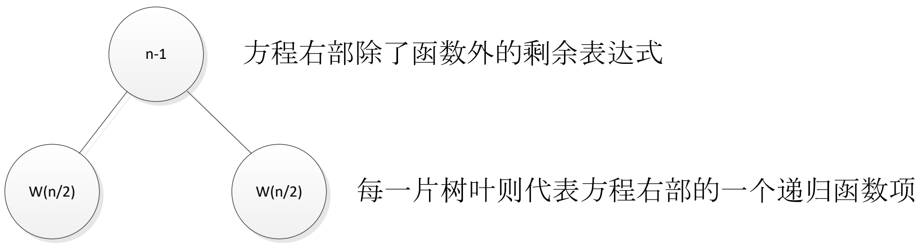
5.2.1. 概念

- 树根（非终结点）：方程右部除了函数外的剩余表达式，如 n-1
- 每一片树叶：方程右部的一个递归函数项，如 W(n/2)

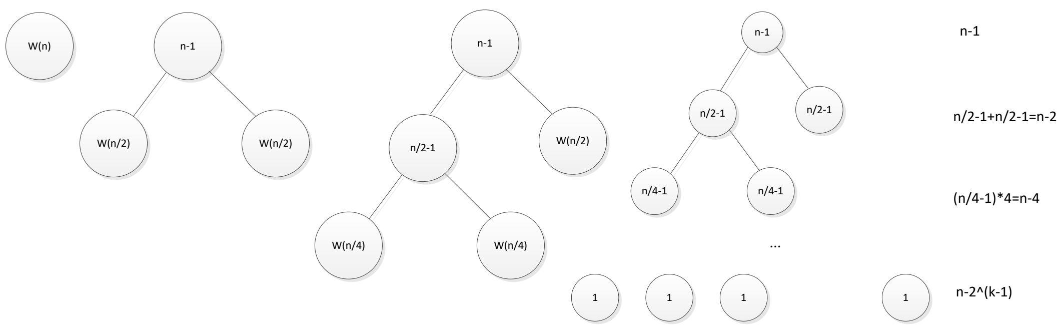
递推方程示例：

$$\begin{cases} W(n) = 2W\left(\frac{n}{2}\right) + n - 1, n = 2^k \\ W(1) = 0 \end{cases}$$

表示为递归树：



递归树分析过程：



（1）每层：每层节点的权值相加，分别获得$n-1$，$\mathrm{n}-2$，$\mathrm{n}-4$，…，$\mathrm{n}-2^{k-1}$

（2）各层：将各层的权值相加，得

$n - 1 + \mathrm{n} - 2 + \mathrm{n} - 4 + \cdots + \mathrm{n} - 2^{k-1}$

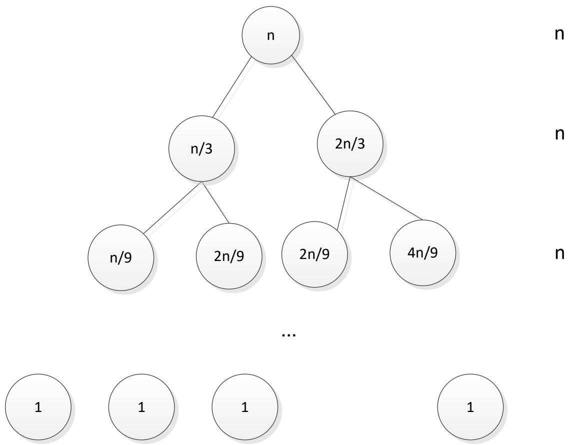
$= nk - (1 + 2 + 4 + \cdots + 2^{k-1})$
$= nk - (2^k - 1)$，由已知得$n = 2^k$
$= nlogn - n + 1$

5.2.2. 示例

例 1.15 求解递推方程

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

（1） 递归树



（2） 树的层数：

最坏情况，考虑最长路径，每次减少为原来的 2/3，为最右分支路径。

不妨设最右分支路径每个节点通项为$(\frac{2}{3})^k n$，k 为层次数，初始值为 0。由于该路径的叶子值为 1，得：

$$(\frac{2}{3})^k n = 1$$
$$\Rightarrow n = (\frac{3}{2})^k$$
$$\Rightarrow k = \log_{3/2} n$$

该树共$\log_{3/2} n$层；

（3） 每层节点数：

每层节点数为 O(n)；

（4） 复杂度，各层权数相加

$$T(n) = O(n \log_{3/2} n) = O(n \log n)$$

5.3. 尝试法

例 1.20 求解下述递推方程

$$\begin{cases} T(n) = 2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + n \\ T(1) = 1 \end{cases}$$

解：估计递推方程的解为$T(n)=O(cnlogn)$，即**存在**$c>0,n_0$,使得当$n\geq n_0$时，有$T(n)\leq cnlogn(n\geq 2)$，进行归纳证明。

（1）当$n=2$有$T(2)=2\mathrm{T}(1)+2=4, \mathrm{c}2\log 2=2\mathrm{c}$,只要$\mathrm{c}=2$就有$T(2)\leq c2\log_2 2$
（2）假设对于一切小于 n 的自然数 k，$T(k)\leq cklogk$成立，那么有：

$$T(n)$$
$$=2T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+n$$
$$\leq 2T\left(\frac{n}{2}\right)+n$$ T(n)为复杂度函数，随着规模增大，时间必然增大，因此为增函数。
$$\leq 2\left(c\frac{n}{2}\log\frac{n}{2}\right)+n$$,由归纳假设得。
$$=cn(\log n-\log 2)+n$$
$$=cn(\log n-1)+n$$
$$=cnlogn-cn+n$$
$$=cnlogn-n(c-1)$$，当$(c>1)$时
$$\leq cnlogn$$

对于$c>1$，只要取$c=2$，对一切 n 都有$T(n)\leq cnlogn$成立。

5.4. 主定理

定理 1.6 主定理（MasterTheorem）设$\mathrm{a}\geq 1, \mathrm{b}>1$为常数，$f(n)$为函数，T(n)为非负整数，且

$$T(n)=aT\left(\frac{n}{b}\right)+f(n)$$

则有以下结果：

(1) 若$f(n)=O(n^{\log_b a-\varepsilon})$，$\varepsilon>0$，那么$T(n)=\Theta(n^{\log_b a})$
(2) 若$f(n)=\Theta(n^{\log_b a})$，那么$T(n)=\Theta(n^{\log_b a}logn)$
(3) 若$f(n)=\Omega(n^{\log_b a+\varepsilon})$，$\varepsilon>0$，且对于某个常数$\mathrm{c}<1$和所有充分大的 n 有 $\mathrm{a}f(n/b)\leq cf(n)$，那么$T(n)=\Theta(f(n))$

证明：

$$T(n)$$
$$=aT\left(\frac{n}{b}\right)+f(n)$$ $T(n)$第 1 次展开
$$=a(aT\left(\frac{n}{b^2}\right)+f(\frac{n}{b}))+f(n)$$ $T(n)$第 2 次展开
$$=a^2T\left(\frac{n}{b^2}\right)+af(\frac{n}{b})+f(n)$$ $T(n)$第 2 次展开结果整理
$$=a^kT\left(\frac{n}{b^k}\right)+a^{k-1}f\left(\frac{n}{b^{k-1}}\right)+\cdots+af(\frac{n}{b})+f(n)$$ $T(n)$第 k 次展开

为使得$T(n)$经过 k 次展开后获得$T(1)$，不妨设$n=b^k$

$$=a^kT(1)+a^{k-1}f\left(\frac{n}{b^{k-1}}\right)+\cdots+af(\frac{n}{b})+f(n)$$

$= a^k T(1) + \sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$, $a^k = n^{\log_b a}$ （后证）

$= T(1)\boldsymbol{n^{\log_b a}} + \sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$, $c_1 = T(1)$

$$= \boldsymbol{c_1 n^{\log_b a} + \sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})}$$

其中，$a^k = n^{\log_b a}$的推导过程：

$n = b^k$

$\Rightarrow k = \log_b n$

$\Rightarrow a^k = a^{\log_b n}, a^{\log_b n} = n^{\log_b a}$

$\Rightarrow a^k = n^{\log_b a}$

其中，$a^{\log_b n} = n^{\log_b a}$推导过程：

$a^{\log_b n} = n^{\log_b a}$

$\Leftrightarrow \log_b(a^{\log_b n}) = \log_b(n^{\log_b a})$

$\Leftrightarrow \log_b n \log_b a = \log_b a \log_b n$

（1）第一种情况，$f(n) = O(n^{\log_b a-\varepsilon})$

T(n)

$$= c_1 n^{\log_b a} + \sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$$

$= c_1 n^{\log_b a} + O(\sum_{j=0}^{\log_b n-1} a^j (\frac{n}{b^j})^{\log_b a-\varepsilon})$, 由$n = b^k$得$k = log_b n$

$$= c_1 n^{\log_b a} + O(n^{\log_b a-\varepsilon} \sum_{j=0}^{\log_b n-1} a^j (\frac{1}{b^j})^{\log_b a-\varepsilon})$$

$$= c_1 n^{\log_b a} + O(n^{\log_b a-\varepsilon} \sum_{j=0}^{\log_b n-1} a^j \frac{1}{b^{j*(\log_b a-\varepsilon)}})$$

$$= c_1 n^{\log_b a} + O(n^{\log_b a-\varepsilon} \sum_{j=0}^{\log_b n-1} \frac{a^j}{(b^{(\log_b a-\varepsilon)})^j})$$

$$= c_1 n^{\log_b a} + O\left(n^{\log_b a-\varepsilon} \sum_{j=0}^{\log_b n-1} (\frac{a}{b^{(\log_b a-\varepsilon)}})^j\right)$$ 其中，$\frac{a}{b^{(\log_b a-\varepsilon)}} = \frac{ab^\varepsilon}{b^{(\log_b a)}} = \frac{ab^\varepsilon}{a} = b^\varepsilon$

$= c_1 n^{\log_b a} + O(n^{\log_b a-\varepsilon} \sum_{j=0}^{\log_b n-1} (b^\varepsilon)^j)$ 其中，$\sum_{j=0}^{\log_b n-1} (b^\varepsilon)^j = \frac{1-(b^\varepsilon)^{(\log_b n-1)+1}}{1-b^\varepsilon} = \frac{(b^\varepsilon)^{\log_b n}-1}{b^\varepsilon-1}$

$$= c_1 n^{\log_b a} + O(n^{\log_b a-\varepsilon} \frac{(b^\varepsilon)^{\log_b n}-1}{b^\varepsilon-1})$$

$= c_1 n^{\log_b a} + O(n^{\log_b a - \varepsilon} \frac{\boldsymbol{b^{\log_b n^{\varepsilon}} - 1}}{\boldsymbol{b^{\varepsilon} - 1}})$

$= c_1 n^{\log_b a} + O(n^{\log_b a - \varepsilon} \frac{\boldsymbol{n^{\varepsilon} - 1}}{\boldsymbol{b^{\varepsilon} - 1}})$

其中，$\lim_{n \to \infty} \frac{\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}}{n^{\varepsilon}} = \lim_{n \to \infty} \frac{n^{\varepsilon}-1}{n^{\varepsilon}(b^{\varepsilon}-1)} = \lim_{n \to \infty} \frac{1}{b^{\varepsilon}} = b^{\varepsilon}$, 则 $\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1} = \Theta(n^{\varepsilon})$

$\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1} = \Theta(n^{\varepsilon})$，$\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1} = \Theta\left(\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}\right)$

$\Rightarrow \Theta\left(\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}\right) = \Theta(n^{\varepsilon})$

$\Rightarrow O\left(\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}\right) = O(n^{\varepsilon})$

$= c_1 n^{\log_b a} + O(n^{\log_b a - \varepsilon} \frac{\boldsymbol{n^{\varepsilon} - 1}}{\boldsymbol{b^{\varepsilon} - 1}})$

$= c_1 n^{\log_b a} + O(n^{\log_b a - \varepsilon} \boldsymbol{n^{\varepsilon}})$

$= c_1 n^{\log_b a} + O(n^{\log_b a})$

得 T(n)= $c_1 n^{\log_b a} + O(n^{\log_b a})$

1) 显然，$c_1 n^{\log_b a} + O\left(n^{\log_b a}\right) \geq c_1 n^{\log_b a}$

2) 又$c_1 n^{\log_b a} + O\left(n^{\log_b a}\right) \leq c_1 n^{\log_b a} + c_2 n^{\log_b a}$，

因此，由Θ定义得：

$$\mathrm{T(n)} = \Theta(n^{\log_b a})$$

（2）第二种情况，$f(n) = \Theta(n^{\log_b a})$

$\mathrm{T(n)}$

$= \mathrm{c_1 n^{\log_b a}} + \sum_{\mathrm{j=0}}^{\mathrm{k-1}} \mathrm{a^j\, f(\frac{n}{b^j})}$

$= \mathrm{c_1 n^{\log_b a}} + \Theta(\sum_{\mathrm{j=0}}^{\mathrm{k-1}} \mathrm{a^j\, (\frac{n}{b^j})}^{\log_b a})$

$= \mathrm{c_1 n^{\log_b a}} + \Theta(n^{\log_b a} \sum_{\mathrm{j=0}}^{\mathrm{k-1}} \mathrm{a^j}\, (\boldsymbol{\frac{1}{b^j}})^{\boldsymbol{\log_b a}})$

$(\frac{1}{\mathrm{b^j}})^{\log_b a} = \left(\frac{1}{(b^j)^{\log_b a}}\right) = \left(\frac{1}{(b^{\log_b a})^j}\right) = \left(\frac{1}{a^j}\right)$

$= \mathrm{c_1 n^{\log_b a}} + \Theta(n^{\log_b a} \sum_{\mathrm{j=0}}^{\mathrm{k-1}} \mathrm{a^j} \left(\boldsymbol{\frac{1}{a^j}}\right))$

$= \mathrm{c_1 n^{\log_b a}} + \Theta(n^{\log_b a} \sum_{\mathrm{j=0}}^{\log_b n - 1} 1)$

$= \mathrm{c_1 n^{\log_b a}} + \Theta(n^{\log_b a}(\log_b n - 1))$

$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \Theta(n^{\log_b a}\log_b n - n^{\log_b a})$

$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \mathrm{c}_2\left(n^{\log_b a}\log_b n - n^{\log_b a}\right)$

其中,

由于 $\lim\limits_{\mathrm{n}\to\infty}\frac{\mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}}+\mathrm{c}_2(n^{\log_b a}\log_b n-n^{\log_b a})}{n^{\log_b a}\log_b n} = \mathrm{c}_2$

因此,$\mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \mathrm{c}_2(n^{\log_b a}\log_b n - n^{\log_b a}) = \Theta(n^{\log_b a}\log_b n)$

(3) 第三种情况,$f(n) = \Omega(n^{\log_b a+\varepsilon})$

$\mathrm{T}(\mathrm{n})$

$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \sum_{\mathrm{j}=0}^{\mathrm{k}-1}\mathrm{a}^\mathrm{j}\,\mathrm{f}(\frac{\mathrm{n}}{\mathrm{b}^\mathrm{j}})$, $\mathrm{a}f(n/b) \le cf(n)$

$$\le \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \sum_{\mathrm{j}=0}^{\mathrm{k}-1}\mathrm{c}^\mathrm{j}\,\mathrm{f}(n)$$

$$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \mathrm{f}(n)\frac{1-c^k}{1-c}$$

$$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \boldsymbol{\mathrm{f}(n)\frac{1-c^{\log_b n}}{1-c}}$$

由于 $\lim\limits_{\mathrm{n}\to\infty}\frac{f(n)\frac{1-c^{\log_b n}}{1-c}}{f(n)} = \lim\limits_{\mathrm{n}\to\infty}\frac{1-c^{\log_b n}}{1-c} = \frac{1}{1-c}, (\mathrm{c} < 1)$

因此 $f(n)\frac{1-c^{\log_b n}}{1-c} = \Theta(f(n))$

$= \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \boldsymbol{\Theta(f(n))}$

(1) $T(n) = c_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \Theta\left(f(n)\right) = \mathbf{c_1n^{\log_b a}} + \mathrm{c}_2 f(n) \le \boldsymbol{\mathrm{c}_3 f(n)} + \mathrm{c}_2 f(n) \le \mathrm{c}f(n)$

(2) $T(n) = c_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \Theta\left(f(n)\right) = \mathrm{c}_1\mathrm{n}^{\log_\mathrm{b}\mathrm{a}} + \mathrm{c}_2 f(n) \ge \mathrm{c}_2 f(n)$

$= \Theta(f(n))$

其中$f(n) = \Omega(n^{\log_b a+\varepsilon})$

$$\Rightarrow \mathbf{c_1n^{\log_b a}} \le \boldsymbol{\mathrm{c}_3 f(n)}$$