目录

湖南工商大学 算法导论 课程教案

授课题目（教学章、节或主题） **课时安排: 2学时**

第二讲：函数的增长 **授课时间 :第二周周一第1、2节**

教学内容（包括基本内容、重点、难点）：

基本内容： （1）渐近记号：渐近上界；渐近下界；渐近紧确界；函数与运行时间。

（2）标准记号：单调性；取整；模运算。

（3）常用函数：多项式；指数函数；对数函数；阶乘；多重函数；斐波那契函数。

教学重点、难点：重点为渐近记号；难点为斯特林公式的应用。

教学媒体的选择：本章使用大数据分析软件Jupyter教学，Jupyter集课件、Python程序运行、HTML网页制作、Pdf文档生成、Latex文档编译于一身，是算法导论课程教学的最佳选择。

板书设计：黑板分为上下两块，第一块基本定义，推导证明以及例子放在第二块。第一块 整个课堂不擦洗，以便学生随时看到算法流程图以及基本算法理论等内容。

课程过程设计： （1）讲解基本算法理论； （2）举例说明； （3）程序设计与编译； （4）对本课堂进行总 结、讨论； （5）布置作业与实验报告

第二讲 函数的增长

算法的输入n非常大的时候，对于算法复杂度的分析就显得尤为重要。虽然有时我们能通过一定的方法得到较为精确的运行时间，但是很多时候，或者说绝大多数时候，我们并不值得去花精力求得多余的精度，因为精确运行时间中的倍增常量和低阶项已经被输入规模本身的影响所支配。

我们需要关心的是输入规模无限增加，在极限中，运行时间是如何随着输入规模增大而增加的。通常来说，在极限情况下渐进地更优的算法在除很小的输入外的所有情况下将是最好的选择。

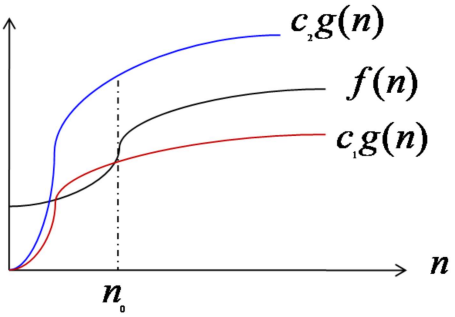
前提假设：本讲定义的所有用在渐近记号中的函数均渐近非负（包括 $f(n), g(n)$）。

一. 渐进记号

1.Θ 记号 (渐近确界)

定义 1：给定一个函数$g(n)$,我们定义$\Theta(g(n))$如下：

$$\Theta(g(n)) = \{f(n) | \exists c_1, c_2, n_0 > 0,\ \forall n > n_0 \text{ 使得 } 0 \le c_1 g(n) \le f(n) \le c_2 g(n)\}$$



记 $f(n) = \Theta(g(n))$ 或 $f(n) \in \Theta(g(n))$.

含义是：若存在常数 c_1 和 c_2，使得对于足够大的n，函数 $f(n)$ 能够"夹入"$c_1 \cdot g(n)$和 $c_2 \cdot g(n)$之间，则 $f(n)$ 属于集合 $\Theta(g(n))$。性质如下：

- $f(n)$ 在一常数因子范围内等于$g(n)$
- $g(n)$ 是 $f(n)$ 的渐近上界和渐近下界
- $g(n)$ 是 $f(n)$ 的渐近紧确界
- Θ定义中，要求$f(n)$ 和 $g(n)$是渐近非负的
- $\Theta(1)$表示算法运行时间与问题的规模无关可写成$\Theta(n^0)$

2.O 记号 (渐近上界)

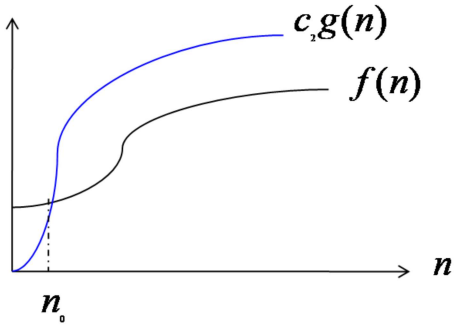
定义 2：给定一个函数 $g(n), Og(n)$表示

$$O(g(n)) = \{f(n)|\exists c, n_0 > 0,\ \forall n > n_0 \text{ 使得 } 0 \leq f(n) \leq cg(n)\}$$

记 $f(n) = O(g(n))$.

含义是：对于足够大的n，$f(n)$ 的值总小于或等于 $c \cdot g(n)$。我们称 $g(n)$ 为 $f(n)$ 的一个渐近上界（asymptotic upper bound）。

下图形象地表示了渐近上界的形式：



3. Ω 记号 (渐近下界)

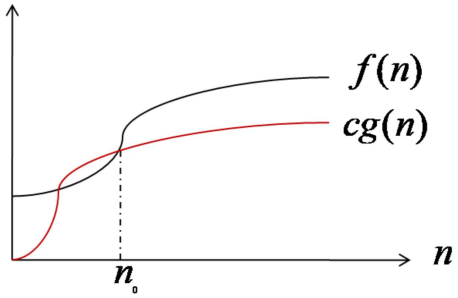
定义 3：给定一个函数 $g(n)$, $\Omega(g(n))$表 示

$$\Omega(g(n)) = \{f(n)|\exists c, n_0 > 0,\ \forall n > n_0 \text{ 使得 } 0 \leq cg(n) \leq f(n)\}$$

记 $f(n) = \Omega(g(n))$.

含义是：对于足够大的n，f(n)的值总大于或等于 $c \cdot g(n)$。我们称 $g(n)$ 为 $f(n)$ 的一个渐近下界（asymptotic lower bound）。

下图形象地表示了渐近下界的形式：



例 1:插入排序最坏运行时间 $T(n) = O(n^2)$, 最好的运行时间 $T(n) = \Omega(n)$

插入排序复杂度分析举例：插入排序最坏情况运行时间的界为 $\Theta(n^2)$（并不表示对每个输入的运行时间界都为$\Theta(n^2)$。首先，插入排序的运行时间为 $O(n^2)$，这意味着最坏的运行时间为 $O(n^2)$。其次，插入排序的运行时间为$\Omega(n)$，这意味着最好的运行时间为$\Omega(n)$（在输入已经排序好的情况下）。

所以，插入排序的运行时间介于 $\Omega(n)$ 和$O(n^2)$ 之间，且是一个尽可能紧确的界，例如，不能说插入排序的运行时间是$\Omega(n^2)$，因为存在一个输入使得排序在 $\Omega(n)$ 内完成，然而这与插入排序最坏运行时间为 $\Omega(n^2)$（由最坏情况运行时间的界为$\Theta(n^2)$ 所蕴涵）并不矛盾，因为存在一个输入使得需要 $\Omega(n^2)$ 的时间。

4. o 记号 (非渐近紧确上界)

定义： 给定一个函数 $g(n), o(g(n))$ 表示

$$o(g(n)) = \{f(n) | \forall c, \exists n_0 > 0, \ \forall n > n_0 \text{ 使得 } 0 \leq f(n) < cg(n)\}$$

记 $f(n) = o(g(n))$,等价于

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

例 2: $2n^2 = O(n^2), 3n = o(n^2), 2n^2 \neq o(n^2)$

n^2是n的上界，成立。

5. ω 记号 (非渐近紧确下界)

定义： 给定一个函数 $g(n)$,$\omega g(n)$表示

$$\omega(g(n)) = \{f(n) | \forall c, \exists n_0 > 0, \ \forall n > n_0 \text{ 使得 } 0 \leq cg(n) < f(n)\}$$

记 $f(n) = \omega(g(n))$,等价于

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

括号里面的是小的就是下界
大的就是上界
比如∞(n^2) 里面是小的，下界

例如： $2n^2 = \omega(n), 2n^2 \neq \omega(n^2)$

n是n^2的下界，成立。

6. 函数间的比较

假定$f(n)$和$g(n)$渐进非负

(1)传递性:

$f(n) = O(g(n))$ 且 $g(n) = O(h(n))$ 蕴含 $f(n) = O(h(n))$

$f(n) = \Omega(g(n))$ 且 $g(n) = \Omega(h(n))$ 蕴含 $f(n) = \Omega(h(n))$

$f(n) = o(g(n))$ 且 $g(n) = o(h(n))$ 蕴含 $f(n) = o(h(n))$

$f(n) = \omega(g(n))$ 且 $g(n) = \omega(h(n))$ 蕴含 $f(n) = \omega(h(n))$

(2)自反性:

$f(n) = \Theta(f(n))$

$f(n) = O(f(n))$

$f(n) = \Omega(f(n))$

(3)对称性:

$f(n) = \Theta(g(n))$ 当且仅当 $g(n) = \Theta(f(n))$

(4)转置对称性:

$f(n) = o(g(n))$ 当且仅当 $g(n) = \omega(f(n))$

$f(n) = O(g(n))$ 当且仅当 $g(n) = \Omega(f(n))$

二. 标准记号与常用函数

1. 向下取整 $\lfloor x \rfloor$ **：** 表示小于等于x的最大整数

2. 向上取整 $\lceil x \rceil$ **：** 表示大于等于x的最小整数

3. 模运算： $a \mod n = a - n\lfloor a/n \rfloor$ ，即商a/n的余数

4. 对数： $\ln n = \log_2 n, ln\, n = \log_e n$

5. 多重对数： $\lg^{(i)} n = \lg(\lg^{(i-1)} n)$

$$\lg^* n = min\left\{i \geqslant 0 : lg^{(i)} n \leq 1\right\}$$

6. 斯特林公式：

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

≈sqrt(2πn)*(n/e)^n

$$\lg(n!) = \Theta(n \lg n)$$

引用及参考：

[1]《Python数据结构与算法分析》布拉德利.米勒等著，人民邮电出版社，2019年9月.

[2] https://www.cnblogs.com/wuwenyan/p/4982713.html (https://www.cnblogs.com/wuwenyan/p/4982713.html)

课后练习

1. 利用斯特林公式证明：$lg(n!) = \Theta(nlg\, n)$

2. 证明：$n! = \omega(2^n)$且$n! = o(n^n)$

2^n < n!
n^n > n!

讨论、思考题、作业： 第35页 练习 3.2.1, 3.2.1

参考资料（含参考书、文献等）：算法导论. Thomas H. Cormen等，机械工业出版社，2017.

授课类型（请打√）：理论课□ 讨论课□ 实验课□ 练习课□ 其他□

教学过程设计（请打√）：复习□ 授新课□ 安排讨论□ 布置作业□

教学方式（请打√）：讲授□ 讨论□ 示教□ 指导□ 其他□

教学资源（请打√）：多媒体□ 模型□ 实物□ 挂图□ 音像□ 其他□

填表说明：1、每项页面大小可自行添减；2、教学内容与讨论、思考题、作业部分可合二为一。